Universidade de Lisboa
Faculdade de Ciências
Departamento de Estatística e Investigação Operacional

Métodos exactos para a resolução de um problema de *phase-in/phase-out* multi-periódico para localização de serviços com capacidades

Margarida Maria da Silva

Dissertação destinada à obtenção do grau de Mestre
em Investigação Operacional

Janeiro, 2005

Universidade de Lisboa
Faculdade de Ciências
Departamento de Estatística e Investigação Operacional

Métodos exactos para a resolução de um problema de *phase-in/phase-out* multi-periódico para localização de serviços com capacidades

Margarida Maria da Silva

Dissertação destinada à obtenção do grau de Mestre
em Investigação Operacional e orientada pelo Professor
Doutor Francisco Saldanha da Gama

Janeiro, 2005

Resumo

O problema analisado nesta dissertação é um problema de *phase-in/phase-out* multi-periódico com restrições de capacidade. Trata-se de um problema de localização caracterizado por considerar no início do horizonte de planeamento, que se assume particionado em vários períodos de tempo, dois conjuntos de localizações, distintos e não vazios. Num deles já existem serviços a operar. No outro podem ser instalados novos serviços. Em cada período pretende-se satisfazer a procura total de um conjunto de clientes de forma a minimizar uma função objectivo, que inclui custos de instalação de novos serviços, custos de remoção de serviços existentes, custos de operação dos serviços e custos de afectação da procura aos serviços que estão em actividade.

Na instalação de um serviço novo é considerado um certo espaço de tempo necessário, por exemplo, para a sua construção. Um tal serviço inicia a sua actividade no início do período seguinte àquele em que é instalado, tendo que se manter operacional até ao final do horizonte de planeamento. Isto significa que no primeiro período não existirão serviços novos em actividade. Se um serviço existente é encerrado em algum dos períodos, finda a sua actividade no final desse período, mantendo-se encerrado até ao final do horizonte de planeamento.

O problema é tratado considerando três abordagens diferentes. É usado um modelo em programação linear inteira mista ao qual são adicionados conjuntos de desigualdades válidas, dando origem a três variantes do modelo básico. Todas as formulações são avaliadas utilizando o procedimento de *Branch-and-Bound* do *general solver IlogCplex 9.0*, com base num conjunto de instâncias geradas aleatoriamente. É, também, proposta uma heurística que permite encontrar um limite superior para o problema. A qualidade das soluções produzidas pela heurística é avaliada tendo em conta a percentagem do desvio relativamente ao valor óptimo. Os limites superiores obtidos são, depois, utilizados como informação adicional para o procedimento de *Branch-and-Bound* acima mencionado. Novamente são apresentados resultados computacionais. Na última abordagem, é proposta e analisada a aplicação do método da decomposição de Benders. São usados cortes obtidos de diferentes formas com o intuito de melhorar a *performance* do método. Os resultados da experiência computacional mostraram que o método é uma boa alternativa ao procedimento de *Branch-and-Bound* do *general solver* na obtenção da solução óptima do problema.

Palavras-chave: Optimização, Localização multi-periódica, Decomposição de Benders.

Abstract

In this thesis a multiperiod *phase-in/phase-out* location problem with capacity constraints is analyzed. In the beginning of a planning horizon that is partitioned into several time periods, two sets of locations are considered. One of them contains locations where facilities are already operating. The other contains (empty) locations where a new facility may be established during the planning horizon. In each period a set of customers should be supplied from the operating facilities so that the total cost is minimized. This cost can be decomposed into establishment, removal, operating (associated with the facilities), and transportation costs (associated with demand satisfaction).

When a new facility is established, some time is required before the facility is available. This time may regard, for instance, construction or remodelation. It is assumed that one time period is enough for this. Accordingly, in the first time period, no new facility is available to supply demand. A new facility always starts operating in the beginning of some time period. Once established, a new facility should be kept operating till the end of the planning horizon.

Existing facilities may be closed during the planning horizon. However, that can only happen in the end of some time period apart from the last one. Actually, if a removal decision were made for the last period, that decision would only be effective after the end of the planning horizon. If an existing facility is removed is some period of the planning horizon, it should be kept closed till the end of the horizon.

The problem is treated considering three different approaches. Firstly, a mixed integer programming formulation is proposed. This formulation is enhanced with valid inequalities as an attempt to strengthen the linear relaxation bound. The performance of the *Branch-and-Bound* procedure imbedded in the general solver *IlogCplex 9.0* when solving the basic model and its variants is evaluated using a set of instances randomly generated. A heuristic procedure to obtain an upper bound to the problem is also proposed. The quality of the solutions produced by the heuristic is evaluated using the same instances as before. The upper bounds provided by this heuristic are then considered as upper cut offs for the *Branch-and-Bound* procedure mentioned above. Again, computational experience is presented. A specific procedure for solving the problem is then proposed. The procedure is actually, a Benders decomposition scheme. Different possibilities for obtaining/enhancing the cuts are proposed/analyzed. The computational experience shows that this method is a good alternative to the *Branch-and-Bound* procedure of the *IlogCplex 9.0* to obtain the optimal solution of the problem.

Keywords: Optimization, Multiperiod location, Benders decomposition.

Agradecimentos

O meu primeiro agradecimento é dirigido ao meu orientador, Professor Doutor Francisco Saldanha da Gama, pela sua assistência incondicional ao longo do tempo para que este trabalho chegasse ao fim. Para isso muito contribuiu o seu empenho e a sua extraordinária capacidade de planeamento, organização, inovação e empenho constantes que me permitiram a aquisição e consolidação de conhecimentos a vários níveis. No plano pessoal, agradeço o apoio em períodos difíceis pelos quais passei e o respeito que sempre demonstrou pelas minhas opiniões mesmo que nem sempre correctas.

À Professora Doutora Maria Eugénia Captivo, gostaria de agradecer a sua atenção sempre que a ela me dirigi e a ajuda que me deu na obtenção de alguma literatura.

Ao meu pai, pessoa que muito admiro, gostaria de lhe agradecer o seu apoio, a todos os níveis, sem o qual dificilmente poderia realizar este trabalho. À minha madrasta, irmãos e especialmente à minha sobrinha Mónica, quero também agradecer todo o apoio que me prestaram.

Aos meus amigos Susana Rosa, Alberto Maia, Gilda Ferreira e Pedro Moura, gostaria de agradecer o seu constante encorajamento e, principalmente, a sua amizade com a qual muito me honram.

À D. Violante Cordeiro, o meu mais profundo agradecimento pelo enorme carinho com que sempre me tratou e pelas palavras certas em horas de desânimo.

O meu último agradecimento é dirigido à pessoa mais importante da minha vida: o meu filho, João André, a quem dedico este trabalho.

Conteúdo

Glossário

c_{ijt}	Custo de satisfação de toda a procura do cliente j no período t, pelo serviço localizado em i.
d_{jt}	Procura do cliente j no período t.
DSP_z	Dual do problema SP_z, também designado por *subproblema dual*.
F_{it}	Custo total associado a um serviço existente (novo) localizado em i, se t for o seu último (primeiro) período de funcionamento.
f_{it}	Custo de operação/manutenção do serviço i no período t.
$\mathcal{F}(P)$	Conjunto das soluções admissíveis do problema P.
g_{it}	Custo de encerramento de um serviço existente na localização i, no final período t.
h_{it}	Custo de instalação de um serviço que vai iniciar a sua actividade na localização i, no início do período t.
HP	Horizonte de planeamento para o qual se pretende um plano de localização.
I	Conjunto de todas as localizações.
I^c	Conjunto de localizações onde existem serviços em funcionamento no início do horizonte de planeamento, também designados por serviços existentes.
$\#I^c$	Número de serviços existentes.
I^o	Conjunto de localizações onde novos serviços poderão ser instalados ao longo do horizonte de planeamento, também designados por serviços novos.

$\#I^o$	Número de serviços novos.
i	Índice que representa um serviço/localização.
J	Conjunto de todos os clientes.
j	Índice que representa um cliente.
$maxt[i]$	Último período de funcionamento de um serviço $i \in I^c$.
$mint[i]$	Primeiro período de funcionamento de um serviço $i \in I^o$.
$m = \#J$	Número de clientes.
$n = \#I$	Número de localizações.
$PLIM$	**P**rogramação **L**inear **I**nteira **M**ista.
$PLMC$	**P**roblema de **L**ocalização **M**ulti-periódico com **C**apacidades.
PM	Problema Mestre.
PM_R	Problema Mestre relaxado (ou restrito).
Q_i	Capacidade do serviço associado à localização i.
R	Subconjunto de $\Lambda\Pi V$.
Rl	Relaxação em programação linear.
SP_z	Subproblema primal (com uma dada configuração fixa).
TP	Conjunto de períodos que constituem o horizonte temporal.
$\#TP$	Número de períodos do horizonte temporal.
t	Índice que representa um período do horizonte de planeamento.
$\mathrm{V}(P)$	Valor óptimo do Problema P.
$\Lambda\Pi V$	Conjunto de todos os pontos extremos de $\mathcal{F}(DSP_z)$.

Capítulo 1

Introdução

Localizar um ou mais serviços/facilidades consiste numa importante decisão que pode ser motivada por variados factores. Alguns desses factores podem ser mais relevantes que outros dependendo de aspectos inerentes ao tipo de serviço a localizar. Por exemplo, localizar um serviço de emergência não é o mesmo que localizar um armazém, um aterro sanitário ou mesmo um aeroporto. A resolução de um problema de localização pretende, de uma forma geral, responder a três questões essenciais: quantos serviços instalar, onde o fazer e qual a melhor forma de afectar os clientes (ou populações) aos serviços instalados.

Ao longo dos anos têm sido desenvolvidos inúmeros modelos matemáticos tentando representar muitos problemas do mundo real. A grande maioria dos problemas pertence a uma classe caracterizada pelo facto de não ser conhecido qualquer algoritmo que resolva esses problemas em tempo *polinomial*: a classe *NP-difícil*. Por esta razão começaram a ser desenvolvidos algoritmos heurísticos através dos quais se obtêm, em geral, soluções satisfatórias num tempo computacional aceitável. Estes algoritmos são úteis, nomeadamente no tratamento de problemas de grande dimensão. Além disso, podem ser um bom auxílio quando se utilizam procedimentos exactos pois podem permitir uma convergência mais rápida para a solução óptima, nomeadamente, se as soluções admissíveis forem de boa qualidade (Labbé *et al.* [32]).

O problema de localização mais simples é caracterizado por ser de natureza estática (não entra em linha de conta com o factor tempo) e determinística (todos os parâmetros associados ao problema são conhecidos e não estão sujeitos a qualquer tipo de incerteza) não tendo qualquer capacidade associada aos serviços. Esta é, talvez, a família de problemas de localização mais estudada. De acordo Labbé *et al.* [32] é atribuída a Alfred Weber a introdução do primeiro modelo de localização numa publicação datada de 1909. O objectivo era localizar um único armazém de forma a minimizar a distância total entre este e os vários clientes a servir. É a partir da década de sessenta, contudo, que o problema de localização começou a merecer maior atenção por parte dos investigadores e durante a qual se fizeram progressos significativos (Owen e Daskin [41]).

Gao e Powell [22] salientam que estes problemas sem capacidades gozam da propriedade de afectação total (existe sempre uma solução óptima onde cada cliente fica afecto a um único serviço), propriedade bastante explorada em algoritmos para estes problemas. Erlenkotter [16], Gao e Powell

[22], Guignard [27] e Beasley [4] são alguns trabalhos de referência em que se aborda o problema de localização simples.

Muitos outros problemas de localização não são mais do que extensões do problema de localização simples, como por exemplo, a família à qual pertencem os problemas da *p-mediana* e da *p-localização*, que resultam de fixar, previamente, o número de serviços a localizar. No primero desses modelos, a partir de um conjunto de pontos dado que representando as comunidades, é seleccionado um subconjunto para a localização dos p serviços (medianas) de forma a minimizar as distâncias entre os pontos e estas (Beasley [4], Christifides e Beasley [10] e Galvão e Santibañez-Gonzalez [20] [21]). No segundo modelo, consideram-se dois conjuntos de pontos, disjuntos, em que um deles representa as comunidades a servir e o outro os potenciais locais para instalação dos p serviços. Pretende-se minimizar o 'custo' total envolvido (Magnanti e Wong [34]). Em particular, se os p serviços a instalar forem, por exemplo, serviços de emergência como hospitais, centros de saúde ou postos de polícia é-se conduzido ao problema de *p-centro*. Neste caso, o objectivo é minimizar a maior das distâncias entre qualquer ponto de procura e o serviço que lhe está mais próximo.

Rapidamente se começa a constatar algumas das limitações dos problemas estáticos, nomeadamente a variação ao longo do tempo de certos parâmetros dos problemas como por exemplo os níveis de procura e, consequentemente, os custos de afectação. Segundo Owen e Daskin [41], Ballou [3] é o impulsionador de um tipo de modelo que considera variações de parâmetros ao longo do tempo. Publicado em 1968, Ballou [3] estuda a localização de um armazém de forma a maximizar o lucro ao longo de um horizonte temporal finito.

Outra extensão dos modelos estáticos e que é influenciada pelo tipo de serviço considerado, tem a ver com a possibilidade de existirem vários critérios ou objectivos, pois as decisões de localização, muitas vezes, não se baseiam, exclusivamente, em aspectos económicos (quando este, porém, é um factor determinante, a decisão é muitas vezes baseada apenas nele). Quando existem diferentes critérios e os mesmos sejam conflituosos, caso se considere a sua relevância, entra-se, então, no campo da análise multiobjectivo onde vários critérios são tidos em consideração. Neste tipo de problemas procura-se identificar não uma mas sim um conjunto de soluções que se designam, usualmente, por soluções eficientes (Alves *et al.* [2]). A localização de serviços semi-obnóxidos é um exemplo onde, necessariamente, vários objectivos são considerados por se tratar da localização de um serviço indesejável pela população mas necessário por razões de ordem ambiental (Ferreira *et al.* [18]). No campo dos problemas de localização multi-periódicos com objectivos múltiplos pode ainda referir-se o trabalho de Melachrinoudis e Min [37].

Nos problemas de natureza multi-periódica, os parâmetros variam ao longo do tempo mas de forma determinística, isto é, a variação é conhecida (pode ser estimada por exemplo através de métodos de previsão). Uma variante a estes problemas é aquela na qual os parâmetros surgem como variáveis aleatórias, seguindo uma determinada distribuição de probabilidades.

Ao longo dos anos os problemas de localização foram adquirindo uma importância cada vez maior, surgindo muitas vezes integrados em problemas mais abrangentes nomeadamente, associados a sistemas de produção-distribuição e a sistemas de telecomunicações.

O estudo de problemas de localização para redes de telecomunicações foi iniciado por Hakimi [29] em 1964 com um problema 1-mediana (Labbé *et al.* [32]). A partir daí, muitos modelos para diferentes sistemas de redes foram surgindo. Nesta área, com o desenvolvimento, nos últimos anos, de novas tecnologias nomeadamente a transmissão sem fio, a área dos problemas de localização ganha um novo fôlego. Novos sistemas de rede surgem onde os terminais (utilizadores) deixam de ter características fixas. As conexões passam a fazer-se não só através de ligações físicas mas também através de ligações sem fio (*ondas de rádio*) e portanto mais frágeis. Por este motivo, os terminais móveis não acedem directamente aos fixos (*Routers*), mas sim através de pontos de acesso fisicamente mais próximos (*Acess Points*). E são estes últimos que se pretendem localizar de forma a minimizar a distância percorrida pela ligação sem fio (Goussevskaia e Mateus [26]). Estes são problemas complexos devido aos diferentes comportamentos de mobilidade dos utilizadores dos terminais móveis, fazendo com que a procura num determinado ponto de acesso varie ao longo do tempo.

Os sistemas de informação geográfica onde os problemas de localização estão muitas vezes integrados (pode consultar-se, a esse título, o site *http://www.lac.inpe.br/%7Elorena/ArsigIndex.html*) e nos quais têm um papel fundamental, são outro exemplo de como é importante resolver problemas de localização de forma eficiente.

De um ponto de vista empresarial, sendo a competitividade uma realidade impossível de ignorar, as empresas são, cada vez mais, desafiadas a operar de forma eficiente, o que constitui, muitas vezes, a única forma de garantir a sua sobrevivência. É essencial que os sistemas estejam em condições de atender tão eficientemente quanto possível às necessidades dos clientes. Muitas empresas enfrentam o desafio através de uma boa gestão do seu sistema de abastecimento. A localização tem neste processo um papel fundamental pois o mesmo deve ser iniciado com uma adequada localização dos serviços (por exemplo, permitindo reduzir significativamente os níveis de *stock*). São vários os modelos propostos na literatura que dão relevância a este problema dos quais Cordeau *et al.* [11], Hinojosa *et al.* [30] e Melo *et al.* [38] [39] são alguns exemplos.

Muitas características importantes dos problemas de localização com que nos deparamos no 'dia-a-dia', não são consideradas ou são-no separadamente por diferentes autores. Algumas dessas lacunas foram preenchidas em Melo *et al.* [38] [39], onde é proposto um modelo que reúne simultaneamente todas as características consideradas em modelos existentes na literatura, incluindo três aspectos que, segundo os autores, raramente são considerados em conjunto: relocalização, oportunidades de investimento e restrições orçamentais. O problema que se propõe tratar nesta tese, pode, na verdade, ser considerado como caso particular do modelo apresentado no referido trabalho, inserindo-se na classe 1 (*single echelon*) definida pelos autores.

No problema que vai ser estudado neste trabalho considera-se um horizonte de planeamento dividido em vários períodos de tempo e um conjunto de possíveis localizações para os serviços, em que algumas já têm serviços em actividade no início do horizonte de planeamento e nas restantes localizações podem ser instalados novos serviços. Pretende-se determinar quais as localizações que devem ter serviços em funcionamento em cada um dos períodos de forma a satisfazer, na totalidade, todos os pontos de procura, minimizando o custo total. O problema definido nestas condições, e seguindo a terminologia proposta por Roodman e Schwarz [45], pode ser designado por um problema de *phase-in/phase-out* multi-periódico para localização de serviços. Em particular, será assumido que nenhum serviço poderá ter o seu estado inicial alterado mais do que uma vez ao longo do horizonte de planeamento, pelo que, tomada a decisão de instalação de um serviço novo num dado período, o mesmo terá que permanecer em actividade nos períodos de tempo seguintes. Se se tratar de um serviço existente no início do horizonte de planeamento, uma vez decidido o seu encerramento, o mesmo deverá manter esse estado até ao final do horizonte de planeamento. Assumir-se-á ainda que o início da actividade de um serviço novo ocorre sempre no início de um período, que não o primeiro pois considera-se a utilização de um espaço de tempo para a instalação do serviço. Por outro lado, quando um serviço é encerrado num dado período, assumir-se-á que finda a sua actividade no final desse mesmo período. Por estas razões não serão considerados encerramentos de serviços no último período do horizonte de planeamento (as consequências teriam efeito apenas após o horizonte de planeamento) nem instalações no primeiro período tendo em consideração o tempo necessário para a sua instalação, podendo depender, por exemplo, da sua construção. Assim, no primeiro período só estarão disponíveis serviços em localizações nas quais já existiam serviços instalados no início do horizonte de planeamento.

Um dos aspectos relevantes em problemas de difícil resolução, como os prolemas de localização, está relacionado com a dimensão dos mesmos. Saul Gass [23], salienta este facto como estando directamente relacionado com quase todas as dificuldades que surgem, referindo que o campo da programação matemática em sistemas de larga escala será sempre um desafio requerendo continuamente novos desenvolvimentos. Tendo em conta esta ideia, para resolução do problema que vai ser estudado, propõe-se a aplicação de um método de decomposição como alternativa ao clássico procedimento geral de *Branch-and-Bound*. Concretamente, estudar-se-á a aplicação da técnica da decomposição de Benders.

No capítulo 2 far-se-á uma breve síntese de alguma da literatura existente dedicada aos problemas de localização de natureza multi-periódica.

No capítulo 3 será descrito o problema em estudo nesta tese com maior detalhe e serão apresentadas quatro formulações equivalentes para o problema sendo três delas nada mais do que a formulação base com mais algumas desigualdades válidas. Apresentar-se-ão ainda os resultados obtidos a partir da resolução desses modelos utilizando o procedimento de *Branch-and-Bound* do *IlogCplex 9.0*. Os testes são efectuados considerando um conjunto de instâncias geradas aleatoriamente. Ainda no capítulo 3, será proposto um procedimento heurístico com o intuito de obter uma solução admissível para o problema. A avaliação do seu desempenho será feita com base no tempo de *CPU* e na percentagem dos desvios relativamente ao valor óptimo do problema quando consideradas as mesmas instâcias usadas atrás. Os limites superiores obtidos serão usados como *upper-cut-off* inicial no procedimento de *Branch-and-Bound* do *Ilog Cplex 9.0* [13]. Novamente apresentam-se e analisam-se os resultados.

O capítulo 4 é o capítulo principal desta dissertação e propõe a aplicação do método da decomposição de Benders como alternativa para resolver o problema de *phase-in/phase-out* detalhado no capítulo 3. Serão apresentadas diferentes formas de obtenção de cortes de Benders e far-se-á uma comparação dos resultados obtidos a partir da aplicação de cada uma das estratégias. Novamente é apresentada experiência computacional obtida a partir das instâncias acima mencionadas.

O capítulo 5 será dedicado à análise de resultados, conclusões e propostas de trabalho futuro.

Capítulo 2

Resenha histórica

Neste capítulo faz-se uma breve síntese de alguma da literatura existente no campo dos problemas de localização multi-periódicos.

Uma grande parte da literatura existente na área da localização de serviços/facilidades, centra-se nos problemas de natureza estática e determinística. Isto significa que a instalação de serviços acontece num único instante possível e que nenhum parâmetro do problema está sujeito a qualquer tipo de incerteza. No entanto, esta literatura revela-se insuficiente para muitos dos problemas que surgem no 'dia-a-dia'. Segundo Owen e Daskin [41], o primeiro trabalho a reconhecer as limitações dos modelos estáticos de localização foi publicado em 1968 por Ballou [3] no qual se propõe localizar um armazém maximizando o lucro ao longo de um horizonte temporal finito e particionado em vários períodos de tempo.

Erlenkotter [17] aponta duas razões essenciais para que se deva considerar um modelo multi-periódico/dinâmico: a variação da procura e/ou custos ao longo do tempo e a existência de custos significativos para relocalização e/ou redimensionamento dos serviços. Nestes casos, de acordo com o autor, deve considerar-se um horizonte temporal discreto (isto é, particionado em diversos períodos de tempo) e as decisões deverão ser tomadas para todo esse espaço de tempo.

Ainda de acordo com Owen e Daskin [41], as decisões de localização devem ter em conta não só o estado actual do sistema mas também possíveis alterações a que possa estar sujeito um determinado serviço enquanto estiver em actividade, pelo que, mais uma vez, se torna essencial considerar um horizonte de planeamento construindo um modelo multi-periódico ou dinâmico.

Roodman e Schwarz [44] propõem um algoritmo de *Branch-and-Bound* para um problema de localização multi-periódico sem limitações de capacidade em que a procura varia ao longo do tempo de forma decrescente. É considerado um conjunto de serviços a operar no início de um horizonte de planeamento que se considera finito e discreto. Pretende-se um plano de encerramento (que os autores também designam por estratégia de *phase-out*) de alguns ou todos os serviços e a re-afectação da procura (em cada um dos períodos) aos serviços que permanecem em actividade, de forma a minimizar o custo total. Este custo inclui custos de operação e de encerramento e, ainda, custos de transporte entre os serviços e os pontos de procura. Os serviços encerrados em algum dos períodos, permanecem encerrados até ao final do horizonte de planeamento. O algoritmo descrito para o plano de encerramento tem por base duas propriedades: uma é a propriedade de afectação total já mencionada no capítulo 1 e que se baseia no facto de, não existindo capacidades associadas aos serviços, haver pelo menos uma solução óptima em que cada cliente fica afecto a um só serviço; a outra tem a ver com o facto de que se se relaxar o conjunto de restrições de ligação entre dois períodos consecutivos o problema se decompõe em tantos problemas de programação linear quantos os períodos que constituem o horizonte de planeamento. Estes problemas podem ser facilmente resolvidos por inspecção.

Em 1977, Roodman e Schwarz [45], generalizam o problema considerando uma estratégia de *phase-in/phase-out*. Neste caso, pretende-se um plano para encerramento de serviços existentes (*phase-out*), abrindo novos serviços (*phase-in*) num conjunto de potenciais locais existentes para o efeito. O modelo de *phase-out* que tinha sido proposto anteriormente é adaptado considerando novas restrições entre as quais as que especificam que um novo serviço pode ser aberto em qualquer período mas que uma vez aberto não pode ser encerrado. Os custos de instalação são incluídos na função objectivo.

Erlenkotter [17] faz uma comparação de várias abordagens ao problema de localização multi-periódico. Apresenta, primeiramente, a forma de um modelo básico de localização de natureza multi-periódica considerando, seguidamente, algumas variantes abordadas por diferentes autores. Destas, seleciona um conjunto, a partir das quais faz uma comparação detalhada, avaliando a *performance* de cada abordagem com base em instâncias já utilizadas na literatura. Esboça algumas estratégias e propõe algumas combinações de várias abordagens que conduzem a uma maior eficiência do que quando consideradas separadamente.

Em 1986, Van Roy e Erlenkotter [50], estudam o problema de localização multi-periódico apresentado por Roodman e Schwarz [45] propondo, para a sua resolução, um procedimento de *Branch-and-Bound* que utiliza limites inferiores obtidos por uma heurística dual ascendente (*DYNALOC*) que pode ser vista como uma extensão, ao caso multi-periódico, de uma heurística apresentada por Erlenkotter [16] para o problema estático de localização simples (*DUALOC*). Os autores propõem, ainda, a aplicação do procedimento a outras variantes do problema base analisado. De entre estas, é de realçar a que considera restrições na capacidade dos serviços. Para estas variantes não são apresentados resultados computacionais.

Shulman [47] analisa um problema de localização assumindo um horizonte temporal discreto, no início do qual existe um conjunto de localizações com serviços em actividade. O tipo de serviço em causa é modular, existindo diferentes tipos de módulos cada um com uma determinada capacidade/dimensão. Em cada localização pode ser instalado mais do que um módulo. Assim, ao longo do tempo, pode proceder-se a uma expansão da capacidade de cada localização o que será feito de forma discreta/modular. O autor propõe um algoritmo baseado na relaxação lagrangeana. Os resultados obtidos (a maior instância testada tem 62 localizações, 62 pontos de procura e 10 períodos de tempo) permitiram ao autor considerar eficiente o procedimento apresentado e, segundo o mesmo, pode ser utilizado para problemas de localização de larga escala.

Chardaire *et al.* [9], apresentam metodologias para obter limites, superior e inferior, para um problema de localização multi-periódico sem capacidades. Este problema é semelhante ao problema estudado por Van Roy e Erlenkotter [50] e Roodman e Schwarz [45], mas com maior flexibilidade, permitindo o encerramento de serviços instalados durante o horizonte de planeamento. Não tem em conta custos relacionados com encerramento de serviços. Os autores apresentam uma formulação em programação quadrática mista e propõem uma relaxação lagrangeana das restrições de satisfação da procura. O problema resultante é resolvido usando programação dinâmica. Para obtenção de soluções admissíveis é proposto um procedimento de *Simulated Annealing*. De acordo com os autores, as soluções obtidas são de boa qualidade. Como trabalho futuro sugere-se a inclusão de restrições de capacidade.

Hinojosa *et al.* [30], abordam um problema de localização multi-periódico com dois níveis de serviço e a existência de vários tipos de artigo. Existem limitações na capacidade máxima dos armazéns e dos serviços. No início do horizonte de planeamento existem serviços e armazéns em funcionamento e que podem ser encerrados ao longo do tempo. Existe também um conjunto de localizações onde novos serviços e armazéns podem ser instalados em algums dos períodos. A configuração destes não pode ser alterada mais do que uma vez ao longo do horizonte de planeamento. São apresentadas duas formulações equivalentes em programação linear inteira mista e é proposta a obtenção de limites inferiores através da relaxação lagrangeana das restrições de satisfação da procura e também das restrições associadas à capacidade dos armazéns. São apresentados resultados com base em instâncias geradas aleatoriamente.

Saldanha da Gama [46], aborda, entre outras, a variante do problema de localização multi-periódico em que existem capacidades. No problema analisado assume-se que no início do horizonte de planeamento (finito), existe um conjunto de localizações com serviços em actividade e, também, um outro conjunto onde é possível instalar novos serviços ao longo do horizonte de planeamento com a particularidade, à semelhança do que acontecia com o modelo proposto por outros autores já mencionados, de que um serviço aberto em algum dos períodos do horizonte de planeamento, assim deve permanecer até ao final do mesmo e se algum dos serviços inicialmente em actividade for encerrado em algum dos períodos (com excepção do último período) deve permanecer fechado até ao final do horizonte de planeamento. Nesse trabalho procura-se minimizar o custo total, o qual inclui custos de operação, instalação, encerramento e custos de transporte entre os serviços e os clientes. Na obtenção de limites inferiores para o valor óptimo, é proposto o uso de uma heurística dual ascendente, tendo por base o trabalho desenvolvido por Guignard e Spielberg [28] para o caso estático. Saldanha da Gama [46] estuda, ainda, uma relaxação Lagrangeana do conjunto de restrições associado à limitação da capacidade dos serviços, usando como vector de multiplicadores inicial na optimização do dual lagrangeano, o valor obtido para as variáveis duais associadas a este conjunto de restrições pela heurística dual ascendente. Ao relaxar este conjunto de restrições obtém-se um problema de localização multi-periódico que o autor designa por problema de localização multi-periódico com custos generalizados. Para obtenção de limites superiores é sugerida uma heurística que parte de uma solução que pode ser admissível ou não (sendo no último caso obtida uma solução admissível a partir desta), aplicando-se de seguida um algoritmo de pesquisa local que procura uma configuração que conduza a uma solução de custo inferior.

Ainda em Saldanha da Gama [46] é apresentado um modelo no qual são incluídas restrições orçamentais mas que não tem em consideração limitações na capacidade dos serviços. Neste caso na função objectivo são considerados apenas custos de operação associados aos serviços, sendo incluídas no modelo as restrições que limitam o investimento (em cada período) com o encerramento e com a instalação de serviços. É proposta uma relaxação lagrangeana das restrições orçamentais, ficando para resolver, mais uma vez, um problema de localização multi-periódico com custos generalizados.

Dias *et al.* [14] consideram um problema de localização sem capacidades no qual os serviços existentes no início do horizonte de planeamento podem ser encerrados em algum dos períodos mas permitindo a sua reabertura (na mesma localização) mais à frente. Os custos de reabertura diferem dos custos de abertura. Existe ainda um conjunto de localizações onde novos serviços podem ser instalados em qualquer um dos períodos do horizonte de planeamento. É proposta uma heurística baseada no procedimento dual ascendente de Van Roy e Erlenkotter [50], sendo o processo complementado com a utilização de um procedimento de *Branch-and-Bound* para a determinação da solução óptima. No entanto, não são apresentados resultados de testes computacionais. Propõem como trabalho futuro uma generalização, considerando restrições na capacidade dos serviços. Esta variante é então considerada em Dias *et al.* [15] onde são considerados alguns modelos que envolvem capacidades associadas aos serviços, que se consideram constantes ao longo do horizonte de planeamento. Uma variante tem em conta limitações da capacidade máxima de cada serviço; na outra impõem-se limitações, quer na capacidade máxima quer na capacidade mínima. É, ainda, apresentado um modelo em que a utilização de um serviço aumenta ao longo do tempo. Permite-se que um serviço seja encerrado sem que a sua capacidade tenha sido esgotada. Se houver lugar a reabertura, a capacidade remanescente é adicionada à capacidade associada ao serviço. São apresentadas heurísticas para obtenção de soluções admissíveis para cada uma das variantes do problema consideradas, as quais se baseiam no trabalho de Van Roy e Erlenkotter [50] e em alguns resultados apresentados em Saldanha da Gama [46]. Embora não sejam exibidos resultados, os autores referem que a heurística apresentada conduz a soluções que consideram de boa qualidade, embora não tenha sido dado a conhecer as dimensões das instâncias sujeitas a teste. Referem ainda que os limites inferiores obtidos considerando a melhor solução dual encontrada situam-se longe do valor óptimo. Os autores sugerem como trabalho futuro uma reformulação dos modelos apresentados considerando que toda a procura associada a um cliente seja satisfeita por apenas um serviço (afectação total).

Na classe dos problemas em que se limita o número de serviços que operam em cada um dos períodos do horizonte de planeamento, os problemas de p-mediana de natureza multi-periódica, foram, segundo Galvão [19], introduzidos por Wesolowsky e Truscott [51]. Neste trabalho considera-se que um serviço pode ser encerrado em qualquer um dos períodos do horizonte de planeamento mas, em qualquer um dos períodos, têm que estar em funcionamento exactamente p serviços (número igual para todos os períodos). Na função de minimização de custos têm em conta custos de abertura e encerramento mas não consideram custos de operação. Em 1990, Galvão e Santibañez-Gonzalez [20] generalizam o problema considerando um número limitado de serviços em cada período mas variável de período para período, ou seja, consideram um problema que denominam de p_k-mediana. São considerados apenas custos de abertura e transporte, não são tidos em conta custos de operação e encerramento de serviços. É proposta a obtenção de um limite inferior para o problema através da resolução de diversos problemas de p-mediana (estáticos), mais concretamente, um para cada período do horizonte de planeamento. Para obtenção de uma solução admissível e, consequentemente, de um limite superior, é proposto um algoritmo baseado em programação dinâmica. São apresentados resultados para um conjunto de instâncias geradas aleatoriamente que atingem os 70 clientes e 70 localizações com um máximo de 8 períodos de tempo. É, ainda, sugerido o desenvolvimento de um método exacto para resolver o problema. Galvão e Santibañez-Gonzalez [21] consideram o mesmo problema, propondo uma heurística lagrangeana para obtenção de soluções admissíveis. O procedimento é testado com base em duas relaxações, nomeadamente a relaxação das restrições de satisfação da procura e a relaxação das restrições de afectação. Usando um conjunto de instâncias de teste que abrangem dimensões da ordem dos 50 clientes, 50 localizações e 7 períodos de tempo, conclui-se que o uso da primeira produz melhores resultados que a segunda, quer em relação aos limites obtidos quer em relação aos tempos de *CPU*. Comparativamente aos resultados apresentados em Galvão e Santibañez-Gonzalez [20], observaram-se melhorias significativas em relação aos limites inferiores e tempos de *CPU*, mas apenas muito ligeiras nos limites superiores.

Melo *et al.* [38] [39], tal como mencionado no capítulo 1, apresentam um modelo que abrange simultaneamente muitas variantes consideradas na literatura mas tratadas separadamente. Uma grande maioria dos problemas na área da localização de serviços/facilidades da literatura podem ser considerados casos particulares deste. Na experiência computacional efectuada, foi utlizado o procedimento de *Branch-and-Bound* de um *general solver* para resolver um conjunto de 95 instâncias geradas aleatoriamente. Mediante os resultados obtidos, os autores realçam a necessidade de desenvolver métodos eficientes sugerindo a aplicação de métodos de decomposição para obtenção da solução óptima ou o uso de meta-heurísticas para obtenção de soluções admissíveis. No sentido desta última sugestão surge o trabalho de Melo e Velásquez [40] no qual é apresentada uma heurística que é iniciada com a obtenção de uma solução admissível seguida de uma pesquisa local que se revelou eficiente face aos resultados obtidos em Melo *et al.* [38] [39].

Capítulo 3

Problema de *phase-in/phase-out* multi-periódico para localização de serviços com capacidades

3.1 Introdução

Neste capítulo dar-se-á início ao estudo do problema central desta dissertação: um problema de localização multi-periódico com restrições de capacidade. Concretamente, trata-se de um problema de localização caracterizado por uma variação, ao longo do tempo, de alguns parâmetros, nomeadamente custos e níveis de procura e, ainda, pela existência de uma limitação da capacidade dos serviços.

Considerar-se-á um horizonte de planeamento particionado em vários períodos de tempo. O objectivo é a minimização do custo total envolvido ao longo de todo o horizonte de planeamento, tendo em conta que todos os clientes/populações têm que ser satisfeitos na totalidade em cada um dos períodos.

Vamos considerar a situação em que no início do horizonte de planeamento se tem um determinado conjunto de localizações, potenciais para a instalação de um novo serviço. Considerar-se-á, ainda, um conjunto de localizações, disjunto do anterior, em cada uma das quais se assume existir actualmente, um serviço em actividade. Considerando que nenhum dos conjuntos anteriores é vazio, ter-se-á, para além de eventuais decisões de instalação de novos serviços, eventuais decisões de encerramento de serviços existentes. Seguindo alguma terminologia proposta por Roodman e Schwarz [45] um problema com estas características poder-se-á designar por um problema de *phase-in/phase-out* de localização. Em particular, se no início do horizonte de planeamento não existissem serviços em actividade, estar-se-ia perante um problema de *phase-in*. Por outro lado, se todas as localizações tivessem um serviço em funcionamento, tratar-se-ia de um problema de *phase-out*.

Em termos de complexidade computacional, o problema de localização descrito atrás é um problema *NP-difícil*. De facto, admite como caso particular o problema de localização simples multi-periódico estudado no capítulo 3 de Saldanha da Gama [46], que é um problema também pertencente a esta classe. Desta forma, não é conhecido qualquer algoritmo que resolva o problema em tempo *polinomial*. Este facto não invalida, contudo, a tentativa de usar algoritmos gerais (por meio de um *general solver*) ou mesmo de delinear algoritmos específicos para o problema capazes de obter a sua solução óptima. Neste capítulo analisa-se a possibilidade de resolver o problema recorrendo a um procedimento de *Branch-and-Bound* incluído no *general solver IlogCplex 9.0* [13]. No capítulo 4, delinear-se-á um algoritmo específico para o problema.

Na secção 3.2 depois de serem apresentados alguns pressupostos em relação ao problema e alguma notação a ser usada ao longo da tese, será proposta uma formulação em programação linear inteira mista. Com o objectivo de obter um modelo mais forte em termos de limite inferior obtido por relaxação em programação linear, apresentar-se-ão diferentes conjuntos de desigualdades válidas que não são mais do que adaptações para o modelo multi-periódico, de desigualdades já conhecidas para o problema estático e que são usualmente conhecidas como desigualdades válidas triviais para o problema de localização com capacidades (Leung e Magnanti [33]). As diferentes formulações obtidas são estudadas utilizando o *general solver IlogCplex 9.0* [13]. Procura-se compará-las não só em termos do tempo de *CPU* necessário para a obtenção da solução óptima usando o procedimento de *Branch-and-Bound* incluído nesse *solver*, mas também em termos da qualidade do limite inferior obtido por relaxação em programação linear. A experiência computacional é efectuada usando um conjunto de instâncias aleatoriamente geradas.

Na secção 3.3 será proposta uma heurística para a determinação de uma solução admissível para o problema. O valor desta solução será usado como *upper-cut-off* inicial aquando da utilização do procedimento de *Branch-and-Bound* do *IlogCplex 9.0* [13]. Novamente são apresentados e analisados os resultados de testes computacionais realizados sobre as instâncias mencionadas acima.

Termina-se o capítulo com algumas conclusões e considerações gerais.

3.2 Formulação do problema

3.2.1 Descrição do problema

Como foi mencionado, o problema em estudo neste capítulo é um problema de *phase-in/phase-out* multi-periódico para localização de serviços com restrições de capacidade. O objectivo é determinar, para cada período, quais as localizações que devem ter serviços em actividade e de que forma a procura de cada cliente deverá ser satisfeita a partir dos serviços em actividade de modo a minimizar o custo total envolvido.

Considera-se um horizonte de planeamento particionado em vários períodos de tempo, justificado pelo facto de se assumirem variações de alguns parâmetros ao longo do tempo, tais como os custos associados à *instalação de novos serviços/remoção de serviços existentes*, custos de operação/manutenção e ainda, os custos de satisfação da procura. No último caso a variação reflete alterações dos níveis de procura ao longo do tempo.

Devido à existência de capacidades associadas aos serviços, cada cliente poderá ter que ser servido por mais do que um serviço em cada período (afectação parcial). Assume-se que em cada local poderá ser instalado, no máximo, um serviço. Considera-se ainda neste problema, a existência de um único tipo de serviço.

Tomada a decisão de instalar um novo serviço, há que ter em conta o tempo necessário para a sua instalação. Este tempo pode ter a ver, por exemplo, com a construção das instalações ou com uma remodelação de instalações já existentes mas associadas a outra actividade. Com base neste argumento, vamos considerar que um período do horizonte de planeamento é suficiente para a instalação de um novo serviço. Este facto vai implicar que no primeiro período só tenhamos disponíveis serviços existentes no início do horizonte de planeamento. Vamos ainda assumir que, instalado um novo serviço para entrar em actividade num dado período, o mesmo ficará disponível desde o início desse período.

Os serviços que se encontram em actividade no início do horizonte de planeamento, podem ser encerrados em algum dos períodos. Vamos assumir que o encerramento de um serviço num dado período, tem efeito no final desse período, o que mantém o serviço ainda disponível durante o referido período. Por este motivo, vamos assumir que não serão considerados encerramentos no último período do horizonte de planeamento, uma vez que os efeitos produzidos por tais encerramentos só teriam influência depois do horizonte de planeamento.

Vamos ainda considerar que se um serviço é encerrado num dado período, o mesmo permanecerá encerrado até ao final do horizonte de planeamento e que uma vez instalado um novo serviço em algum dos períodos, o mesmo ficará em actividade até ao final do horizonte de planeamento.

O estado de um serviço em cada período é caracterizado por *aberto* ou *fechado*. Desigar-se-á por configuração dos serviços, uma especificação do estado de cada serviço em todos os períodos do horizonte de planeamento. Uma configuração dos serviços será admissível se e só se, cada um dos serviços tem o seu estado alterado no máximo uma vez durante todo o horizonte de planeamento e se a capacidade total dos serviços em actividade em cada período é suficiente para satisfazer toda a procura nesse período.

Uma solução para o problema é definida por uma configuração para os serviços e pela forma como os clientes são afectos aos serviços em funcionamento em cada período do horizonte de planeamento. Uma solução é admissível se e só se, a configuração correspondente for admissível e se toda a procura associada a cada cliente é satisfeita na totalidade em todos os períodos do horizonte de planeamento.

Observação *Uma configuração admissível para os serviços define uma solução admissível para o problema dado que, sendo a configuração admissível, em cada período a capacidade operacional é suficiente para satisfazer a procura. Assim, o problema de afectação dos clientes aos serviços em actividade tem sempre solução admissível. De facto, fixados os serviços resta estabelecer a ligação entre cada cliente e os serviços em funcionamento. Dado existir capacidade suficiente para satisfazer a procura total em cada período e podendo os clientes ser servidos por mais do que um serviço, há sempre uma forma de serem satisfeitos na totalidade. Na verdade, este subproblema resume-se à resolução de um problema de transportes para cada período. Em cada um destes problemas, a soma das disponibilidades (capacidade total dos serviços em funcionamento no período em causa) é não inferior à soma da procura (procura total dos clientes no mesmo período).* ♣

Como já foi mencionado anteriormente, o problema atrás descrito pode ser visto como um caso particular do problema mais geral apresentado por Melo *et al.* [39], onde é considerado um modelo para o problema de localização multi-periódico com vários níveis de distribuição e no qual os autores procuram reunir conjuntamente todas as carcterísticas estudadas separadamente por outros autores. O problema em estudo nesta tese corresponde ao problema com um único nível de distribuição.

3.2.2 Notação

Apresenta-se, de seguida, alguma notação que vai ser utilizada ao longo deste trabalho.

Comecemos por definir o horizonte de planeamento, designando por TP, o conjunto de períodos no qual o mesmo se assume particionado.

$$\mathrm{TP} = \{1,\, 2,\, 3,\, \ldots,\, \mathrm{T}\}$$

T designa o último período do horizonte de planeamento e coincide com o número de períodos que o constituem. O índice $t \in TP$, será utilizado para representar um período do horizonte de planeamento.

O conjunto de locais onde inicialmente existem serviços em actividade, será representado por I^c. A decisão de encerramento destes serviços só pode ser tomada em relação aos períodos $t = 1, \ldots, T-1$, uma vez que se assumiu que não são admitidos encerramentos no último período.
O conjunto de potenciais locais para a instalação de novos serviços vai ser representado por I^o. Como foi referido atrás, um serviço novo só pode iniciar a sua actividade num dos períodos $t = 2, \ldots, T$.

O conjunto de todas as localizações é representado por I. Assim, $I = I^c \bigcup I^o$. Recorde-se que I^c e I^o são conjuntos disjuntos. O índice i será usado para representar um serviço/localização.

Assume-se que a capacidade dos serviços é conhecida e, para cada localização, constante ao longo do tempo, considerando-se que o tipo de serviço a instalar não é susceptível de ter a sua capacidade alterada durante o horizonte de planeamento, como escolas ou armazéns que têm limitações de espaço. Q_i representa a capacidade do serviço associado à localização $i \in I$.

Observação *Por simplificação, vamos considerar que sempre que é referido um serviço $i \in I^c$ ou um serviço $i \in I^o$ se está, implicitamente, a referir a localização na qual esse serviço está a funcionar. Estes serviços serão muitas vezes designados, respectivamente, por serviços existentes e serviços novos.* ♣

Tal como se referiu na introdução deste capítulo, cada cliente, em cada período, pode ser servido por um ou mais serviços que estejam em funcionamento. Designar-se-á por J, o conjunto de todos os clientes. Um cliente será representado pelo índice $j \in J$ e a sua procura num determinado período $t \in TP$, será representada por d_{jt}. Assume-se que a procura de cada cliente é sempre não negativa e terá que ser totalmente satisfeita em cada um dos períodos do horizonte de planeamento.

Um aspecto importante tem a ver com a avaliação de uma instância quanto à sua admissibilidade (ou não admissibilidade). Uma instância não será admissível se houver pelo menos um período do horizonte de planeamento em que a capacidade máxima que é possível instalar não seja suficiente para satisfazer toda a procura nesse período. Caso contrário a instância é admissível. Há que ter em conta que no primeiro período só estão disponíveis os serviços existentes.

O algoritmo 3.1 sistematiza a análise de uma instância quanto à sua admissibilidade.

Algorithm 3.1 Teste para verificação da admissibilidade de uma instância

InstAdm = 1
if $\sum_{i \in I^c} Q_i < \sum_{j \in J} d_{j1}$ **then**
  InstAdm = 0
else
  for $t = 2, \ldots, T$ **do**
    if $\sum_{i \in I} Q_i < \sum_{j \in J} d_{jt}$ **then**
      InstAdm = 0. Stop.

No algoritmo 3.1, a variável InstAdm devolve o valor 1 ou 0, respectivamente, se a instância tem ou não solução admissível.

O objectivo do problema que se tem vindo a descrever, é a minimização dos custos envolvidos ao longo de todo o horizonte de planeamento. Em cada período há a considerar os custos de instalação/remoção e os custos de operação associados aos serviços. Há ainda que considerar os custos associados à satisfação da procura.

Considere-se então

$$c_{ijt} = \text{custo de satisfazer na totalidade, a procura do cliente } j\text{, pelo serviço localizado em } i\text{, no período } t,\ i \in I,\ j \in J,\ t \in \mathrm{TP}$$

Para cada $i \in I$ e $t \in \mathrm{TP}$, defina-se

$$f_{it} = \text{custo de operação/manutenção do serviço } i \text{ no período } t$$

Note-se que para os serviços $i \in I^o$, os dois custos anteriores não existem para o período 1.

Para cada $i \in I^c$, $t = 1, \ldots, T-1$ seja

$$g_{it} = \text{custo de remoção do serviço existente em } i \text{ no fim período } t$$

e para cada $i \in I^o$, $t = 2, \ldots, T$ seja

$$h_{it} = \text{custo associado à instalação de um serviço que vai iniciar a sua actividade na localização } i \text{ no início do período } t$$

Por simplificação, vai-se ainda definir:

$$\bar{F}_{it} = \sum_{\tau=1}^{t} f_{i\tau}, \qquad i \in I^c,\ t = 1, \ldots, T$$

e

$$\bar{F}_{it} = \sum_{\tau=t}^{T} f_{i\tau}, \qquad i \in I^o,\ t = 2, \ldots, T$$

Considerar-se-á ainda

$$F_{it} = \bar{F}_{it} + g_{it}, \qquad i \in I^c,\ t = 1, \ldots, T-1 \tag{3.1}$$

e

$$F_{iT} = \bar{F}_{iT}, \qquad i \in I^c \tag{3.2}$$

$$F_{it} = \bar{F}_{it} + h_{it}, \qquad i \in I^o\ t = 2, \ldots, T \tag{3.3}$$

A matriz de custos F_{it}, $i \in I$, $t \in TP$, tornará mais fácil a determinação do custo de uma configuração, pois inclui os custos de instalação, remoção e operação. Assim, para cada $i \in I^c$ e $t = 1, \ldots, T-1$, F_{it} é o custo total associado ao serviço i, se o período t for o seu último período de funcionamento e F_{iT} é o custo total associado ao serviço i se este se mantiver em funcionamento ao longo de todo o horizonte de planeamento. Para cada $i \in I^o$ e $t = 2, \ldots, T$, F_{it} é o custo total associado ao serviço i se o período t for o seu primeiro período de funcionamento.

Assume-se neste trabalho que todos os custos envolvidos são positivos. Contudo, há situações em que faz sentido considerar a possibilidade de os custos de encerramento serem negativos. Saldanha da Gama [46], faz referência a esta possibilidade, justificando com o facto de, por exemplo, se poderem obter ganhos com a venda das instalações (que podem eventualmente cobrir custos positivos inerentes ao encerramento como os custos com re-educação de pessoal).

3.2.3 Modelo

Com o objectivo de formular o problema utilizando um modelo de programação linear inteira mista, vão ser considerados dois tipos de variáveis de decisão:

- Variáveis de localização.
- Variáveis associadas à afectação dos clientes aos serviços.

Mais precisamente, para $i \in I$, $j \in$ J e $t \in TP$ seja

x_{ijt} = fracção da procura do cliente j no período t, satisfeita pelo serviço localizado em i

Para $i \in I^c$ e $t \in TP$, considerar-se-á

$$z_{it} = \begin{cases} 1, & \text{se o serviço } i \text{ se mantiver em funcionamento até ao final} \\ & \text{do período } t, \text{ sendo encerrado nessa altura} \\ 0, & \text{caso contrário} \end{cases} \tag{3.4}$$

com $z_{iT} = 0$.

Para cada $i \in I^o$ e $t \in TP$, considerar-se-á

$$z_{it} = \begin{cases} 1, & \text{se o serviço } i \text{ inicia o seu funcionamento no período } t \\ 0, & \text{caso contrário} \end{cases} \tag{3.5}$$

com $z_{i1} = 0$,

isto é, nenhum serviço existente pode ter o seu estado alterado no último período do horizonte de planeamento. O mesmo sucede com os serviços $i \in I^o$ no primeiro período.

Tendo em conta as variáveis de localização acima definidas, tem-se que um serviço $i \in I^c$ estará em funcionamento no período $t \in TP$, se e só se,

$$\sum_{\tau=1}^{t-1} z_{i\tau} = 0$$

ou seja, se até ao período $t-1$ o seu estado não tiver sido alterado (para fechado).

Por sua vez, um serviço $i \in I^o$, estará em funcionamento no período $t \in TP \setminus \{1\}$, se e só se,

$$\sum_{\tau=2}^{t} z_{i\tau} = 1$$

ou seja, se em algum dos períodos $2, \ldots, t$, o seu estado for alterado (para aberto). Assim, os serviços efectivamente em funcionamento num dado período $t \in TP$, são os que pertencem ao seguinte conjunto:

$$O_t = \left\{ i \in I^c : \sum_{\tau=1}^{t-1} z_{i\tau} = 0 \right\} \cup \left\{ i \in I^o : \sum_{\tau=2}^{t} z_{i\tau} = 1 \right\}$$

Estão agora reunidas as condições para a apresentação da formulação do problema em programação linear inteira mista. O problema de *phase-in/phase-out* multi-periódico para localização de serviços com capacidades pode ser formulado através do seguinte modelo que será designado por $PLMC_1$:

$(PLMC_1)$

$$Min \quad \sum_{i\in I}\sum_{j\in J}\sum_{t\in TP} c_{ijt}x_{ijt} + \sum_{i\in I^c}\sum_{t=1}^{T-1} F_{it}z_{it} + \sum_{i\in I^c} F_{iT}\Big(1-\sum_{\tau=1}^{T-1} z_{i\tau}\Big) + \sum_{i\in I^o}\sum_{t=2}^{T} F_{it}z_{it} \tag{3.6}$$

$s.a:$

$$\sum_{i\in I} x_{ijt} = 1 \qquad j\in J,\ t\in TP \tag{3.7}$$

$$\sum_{j\in J} d_{jt}x_{ijt} \leq Q_i\left(1-\sum_{\tau=1}^{t-1} z_{i\tau}\right) \qquad i\in I^c,\ t\in TP \tag{3.8}$$

$$\sum_{j\in J} d_{jt}x_{ijt} \leq Q_i \sum_{\tau=2}^{t} z_{i\tau} \qquad i\in I^o,\ t\in TP \tag{3.9}$$

$$\sum_{t\in TP} z_{it} \leq 1 \qquad i\in I \tag{3.10}$$

$$0\leq x_{ijt}\leq 1 \qquad i\in I,\ j\in J,\ t\in TP \tag{3.11}$$

$$z_{it}\in\{0,1\} \qquad i\in I,\ t\in TP \tag{3.12}$$

A função objectivo (3.6) representa o custo total ao longo do horizonte de planeamento, o qual envolve custos de satisfação da procura, custos de operação, de instalação e de remoção de serviços. Em particular, a terceira componente da função objectivo, traduz o custo total de operação dos serviços $i \in I^c$ que estiverem em funcionamento ao longo de todo o horizonte de planeamento. O conjunto de restrições (3.7) assegura que a procura de todos os clientes é satisfeita na totalidade em todos os períodos. As restrições (3.8) e (3.9) são restrições que garantem, por um lado, que cada cliente em cada período, será afecto a serviços em actividade, e por outro que a capacidade de cada serviço não é ultrapassada. As restrições (3.10) garantem que cada serviço só tem o seu estado inicial alterado, no máximo, uma vez ao longo do horizonte de planeamento. Finalmente, os conjuntos de restrições (3.11) e (3.12) são restrições de domínio.

Note-se que a condição $0 \leq x_{ijt} \leq 1$, pode ser substituída por $x_{ijt} \geq 0$, uma vez que as restrições (3.7) impedem que cada um dos x_{ijt}, $i \in I$, $j \in J$ e $t \in TP$ (que são não negativos) ultrapasse o valor 1.

O modelo (3.6) - (3.12) é suficiente para traduzir matematicamente o problema em estudo. Contudo, e pensando desde já no tratamento do modelo, é quase inevitável explorar a relaxação em programação linear como forma de obter limites inferiores e também como componente essencial de métodos muito utilizados nomeadamente de *Branch-and-Bound*. A relaxação linear de (3.6) - (3.12) é obtida relaxando as restrições de integralidade, (3.12), ou seja, substituindo no modelo esse conjunto de restrições por

$$0\leq z_{it}\leq 1 \qquad i\in I,\ t\in TP \tag{3.13}$$

Note-se que basta considerar $z_{it} \geq 0$, pois o conjunto (3.10), garante que cada z_{it}, $i \in I$ e $t \in TP$, é menor ou igual a 1.

3.2.4 Variantes do modelo / Desigualdades válidas

Uma primeira forma de obter um modelo eventualmente mais forte em termos de relaxação linear consiste em considerar as restrições

$$x_{ijt} \leq \left(1 - \sum_{\tau=1}^{t-1} z_{i\tau}\right) \qquad i \in I^c,\ j \in J,\ t \in TP \tag{3.14}$$

$$x_{ijt} \leq \sum_{\tau=2}^{t} z_{i\tau} \qquad i \in I^o,\ j \in J,\ t \in TP \tag{3.15}$$

que não são mais do que a adaptação ao caso multi-periódico das desigualdades válidas triviais da formulação dita *forte* para o problema de localização estática (Leung e Magnanti [33]).

Estas restrições asseguram que, em cada período, nenhum cliente será afecto a um serviço que não esteja em actividade. Essa possibilidade já está assegurada pelos conjuntos (3.8) e (3.9), fazendo com que (3.14) e (3.15) se tornem redundantes no problema inteiro misto.

Vamos designar por $PLMC_2$ o modelo que se obtém de $PLMC_1$ acrescentando os conjuntos de restrições (3.14) e (3.15).

Uma desigualdade válida também frequentemente considerada no problema estático de localização com capacidades é a que garante que a capacidade operacional tem que ser suficiente para satisfazer a procura. A adaptação ao problema em estudo nesta tese conduz a

$$\sum_{i \in I^c} Q_i \left(1 - \sum_{\tau=1}^{t-1} z_{i\tau}\right) + \sum_{i \in I^o} Q_i \left(\sum_{\tau=2}^{t} z_{i\tau}\right) \geq \sum_{j \in J} d_{jt} \qquad t \in TP \tag{3.16}$$

Para cada período $t \in TP$, o termo esquerdo de (3.16) representa a capacidade total dos serviços em actividade nesse período e o termo direito a procura total no mesmo período.

Designar-se-ão por $PLMC_3$ e $PLMC_4$, os modelos que resultam da inclusão das restrições (3.16), respectivamente, em $PLMC_1$ e $PLMC_2$.

Em resumo ter-se-á:

$PLMC_1$	$\longrightarrow$	*Formulação inicial*
$PLMC_2$	$\longrightarrow$	$PLMC_1$ + (3.14) + (3.15)
$PLMC_3$	$\longrightarrow$	$PLMC_1$ + (3.16)
$PLMC_4$	$\longrightarrow$	$PLMC_1$ + (3.14) + (3.15) + (3.16)

3.2.5 Experiência computacional

Com objectivo de avaliar a viabilidade de utilizar o procedimento de *Branch-and-Bound* do *general solver IlogCplex 9.0* [13] para resolver o problema em estudo, os modelos $PLMC_1$, $PLMC_2$, $PLMC_3$ e $PLMC_4$ foram codificados utilizando a linguagem de modelação *IlogConcert 2.0* [12]. Para um conjunto de instâncias do problema, geradas aleatoriamente, registou-se então o tempo de *CPU* (em segundos) necessário para resolver o problema utilizando o referido *package*. Os testes foram realizados num computador *Pentium IV* a 2.7 GHz com 1 Gb de memória *Ram*.

Resolveram-se ao todo 54 instâncias, as quais foram geradas aleatoriamente utilizando um procedimento semelhante ao proposto por Saldanha da Gama [46], com os devidos ajustamentos para o problema em estudo. De facto, o problema de localização multi-periódico com capacidades estudado em Saldanha da Gama [46], permite que os serviços existentes possam não estar em funcionamento no primeiro período do horizonte de planeamento e que novos serviços possam entrar em funcionamento nesse período. De resto, os parâmetros associados ao problema são os mesmos que estão a ser considerados neste trabalho pelo que a geração das instâncias se pode fazer de forma análoga. Faz-se de seguida uma breve síntese.

Foram geradas instâncias em que o número de localizações considerado foi de 10, 20 e 50 combinado com 20, 50 e 100 clientes em horizontes de planeamento particionados em 5, 10 e 15 períodos de tempo.

Fez-se variar o número de serviços existentes no início do horizonte temporal, $\#I^c$, em função do número de localizações de acordo com três possibilidades: ($\#I^c < \frac{\#I}{2}$), ($\#I^c = \frac{\#I}{2}$) e ($\#I^c > \frac{\#I}{2}$). Assim, para 10 localizações considerou-se $\# I^c = 3$, 5 e 8, para 20 localizações, $\# I^c = 8$, 10 e 16 e, finalmente, para 50 localizações, $\# I^c = 20$, 25 e 45.

Considerou-se um rectângulo de dimensão 10×20, o qual foi dividido longitudinalmente de forma igual pelo número de localizações, obtendo-se $\#I$ faixas. A cada uma destas faixas foi afectada aleatoriamente uma das localizações, gerando-se, em seguida, aleatoriamente as correspondentes coordenadas.

As capacidades dos serviços (associados às localizações) foram geradas aleatoriamente de acordo com uma distribuição Uniforme no intervalo [100, 300].

Os custos operacionais foram gerados da seguinte forma: para os serviços $i \in I^c$ e para o primeiro período, gerou-se aleatoriamente um número de acordo com uma distribuição Uniforme no intervalo [8, 12.5] e multiplicou-se esse valor pela respectiva capacidade. Para cada um dos períodos seguintes, a capacidade operacional obteve-se de acordo com um acréscimo que variou entre 0% e 10% relativamente ao período anterior (gerado de acordo com uma distribuição Uniforme).

Para os serviços $i \in I^o$, dado que as condições do problema não permitem serviços novos a funcionar no primeiro período, não têm associados custos de operação para esse período. Geraram-se então, para o segundo período, valores de acordo com uma distribuição Uniforme no intervalo [8, 12.5]. Para cada um dos períodos seguintes procedeu-se tal como para os serviços existentes.

Somente os serviços $i \in I^c$ podem ser encerrados e tal apenas pode acontecer num dos períodos $t = 1, \ldots, T-1$. Em relação aos custos de encerramento, estes foram gerados para o primeiro período multiplicando cada uma das capacidades por um valor gerado aleatoriamente de acordo com uma distribuição Uniforme no intervalo [7, 25]. Para cada um dos períodos seguintes, obtiveram-se os respectivos valores acrescentando uma quantidade gerada de acordo com uma distribuição Uniforme e que variou entre 0% e 10% relativamente ao período anterior.

Só os serviços $i \in I^o$ têm associados custos de instalação. Estes custos foram gerados da seguinte forma: para o segundo período e para cada um destes serviços, gerou-se aleatoriamente um valor de acordo com uma distribuição Uniforme no intervalo [9, 27] e multiplicou-se pela respectiva capacidade. Para cada um dos períodos seguintes, obtiveram-se os correspondentes valores a partir do período anterior com um acréscimo escolhido aleatoriamente entre 0% e 10%.

Relativamente à procura de cada cliente, para o primeiro período gerou-se aleatoriamente um valor de acordo com uma distribuição Uniforme, no intervalo $[1, \frac{1}{\#J} \sum_{i \in I} Q_i\ ]$. Nos restantes períodos considerou-se uma variação entre -10% e 10% relativamente ao período anterior (de acordo com uma distribuição Uniforme).

Os custos unitários de satisfação da procura no primeiro período, basearam-se na distância *euclideana* entre cada cliente e cada serviço depois de geradas aleatoriamente as coordenadas dos clientes no mesmo rectângulo 10×20 utilizado para as localizações. Para os períodos subsequentes, o custo total de satisfação da procura é obtido a partir do custo do período anterior acrescido de uma variação aleatória escolhida de acordo com uma distribuição Uniforme entre -10% e 10%.

Para obter o custo de satisfação de toda a procura de cada cliente multiplica-se o custo unitário obtido pela procura gerada para esse período.

As tabelas 3.1 - 3.6 apresentam os resultados obtidos na resolução das instâncias geradas. As 3 primeiras colunas identificam respectivamente, o número de períodos do horizonte de planeamento, o número de clientes e o número de serviços em actividade no início do horizonte de planeamento.

Nas colunas seguintes e para cada um dos modelos é apresentado o tempo de CPU (em segundos) associado ao modelo de programação linear inteira mista ($PLIM$) e o tempo de CPU associado à respectiva relaxação em programação linear (*Rl*). Apresenta-se ainda, o desvio relativo, em percentagem, do limite inferior obtido por relaxação linear. Este valor é definido para cada instância por

$$\frac{vOpt - V_{Rl}}{vOpt} \times 100 \tag{3.17}$$

em que $vOpt$ representa o valor óptimo da instância e V_{Rl} representa o valor óptimo da correspondente relaxação linear.

A tabela 3.1 apresenta os resultados referentes aos modelos $PLMC_1$ e $PLMC_2$, para as instâncias com 10 localizações. A tabela 3.2 apresenta o mesmo tipo de informação mas para os modelos $PLMC_3$ e $PLMC_4$. As tabelas 3.3 e 3.4 apresentam os resultados obtidos para as intâncias com 20 localizações. Por fim, as tabelas 3.5 e 3.6, apresentam os resultados correspondentes às instâncias com 50 localizações.

			$PLMC_1$			$PLMC_2$		
T	$\#J$	$\#I^c$	$PLIM$	Rl	% desvio	$PLIM$	Rl	% desvio
		3	0.05	0.04	0.52	0.09	0.07	0.47
5		5	0.22	0.04	3.10	0.39	0.10	1.65
		8	0.16	0.04	2.35	0.30	0.06	1.44
		3	2.17	0.05	6.00	6.03	0.27	4.50
10	20	5	1.64	0.05	5.00	2.81	0.19	3.97
		8	0.75	0.03	4.00	1.55	0.14	3.24
		3	80.14	0.07	6.20	132.17	0.67	4.00
15		5	668.06	0.07	8.76	1058.55	0.66	5.38
		8	3.88	0.08	2.57	7.91	0.44	1.42
		3	0.09	0.04	1.67	0.5	0.20	0.90
5		5	0.47	0.05	7.70	1.94	0.17	6.70
		8	0.66	0.04	3.28	2.02	0.27	1.80
		3	30.64	0.11	7.50	26.13	0.96	6.20
10	50	5	5.41	0.08	3.57	15.41	3.91	3.00
		8	56.20	0.09	6.54	165.97	1.13	4.40
		3	191.69	0.14	7.20	510.98	6.80	5.00
15		5	214.41	0.18	5.00	285.06	7.85	3.40
		8	707.34	0.16	5.84	1453.91	3.03	3.50
		3	0.66	0.08	3.86	11.89	3.10	2.13
5		5	0.11	0.06	0.08	3.80	0.83	0.04
		8	0.84	0.05	2.30	2.27	0.45	1.20
		3	31.50	0.18	5.50	50.11	6.02	4.13
10	100	5	15.08	0.18	4.67	52.95	6.10	2.65
		8	2.06	0.16	1.60	12.67	3.35	0.38
		3	256.28	0.30	7.30	556.94	29.23	5.00
15		5	513.14	0.41	3.70	1600.95	29.70	1.60
		8	312.06	0.38	4.30	384.17	25.01	3.40

Tabela 3.1: Tempo de CPU (seg) e desvio (%) - 10 localizações.

			$PLMC_3$			$PLMC_4$		
T	$\#J$	$\#I^c$	$PLIM$	Rl	% desvio	$PLIM$	Rl	% desvio
		3	0.03	0.04	0.52	0.03	0.02	0.47
5		5	0.13	0.02	3.10	0.23	0.03	1.65
		8	0.09	0.04	2.35	0.09	0.03	1.44
		3	0.91	0.03	6.00	1.58	0.16	4.50
10	20	5	0.69	0.02	5.00	1.87	0.08	3.97
		8	0.69	0.04	4.00	1.58	0.08	3.24
		3	7.24	0.05	6.20	10.20	0.47	4.00
15		5	543.67	0.07	8.76	592.08	0.41	5.38
		8	0.86	0.07	2.57	1.67	0.27	1.42
		3	0.06	0.05	1.67	0.14	0.11	0.90
5		5	0.30	0.05	7.70	0.72	0.09	6.70
		8	0.64	0.05	3.28	1.56	0.14	1.80
		3	6.11	0.07	7.50	8.83	0.58	6.20
10	50	5	0.67	0.06	3.57	1.59	0.47	3.00
		8	6.44	0.08	6.54	9.53	0.75	4.40
		3	84.75	0.13	7.20	67.11	1.40	5.00
15		5	90.42	0.12	5.00	86.95	1.61	3.40
		8	295.91	0.12	5.84	280.11	1.85	3.50
		3	0.39	0.08	3.86	0.98	0.36	2.13
5		5	0.08	0.06	0.08	0.33	0.24	0.04
		8	0.30	0.06	2.30	1.22	0.35	1.20
		3	13.83	0.14	5.50	26.67	0.79	4.13
10	100	5	4.83	0.13	4.67	12.81	1.86	2.65
		8	1.16	0.13	1.60	4.22	0.92	0.38
		3	104.05	0.20	7.30	164.20	5.42	5.00
15		5	318.80	0.28	3.70	240.69	7.00	1.60
		8	115.58	0.22	4.30	133.97	2.08	3.40

Tabela 3.2: Tempo de *CPU* (seg) e desvio (%) - 10 localizações.

			$PLMC_1$			$PLMC_2$		
T	#J	#I^c	$PLIM$	Rl	% desvio	$PLIM$	Rl	% desvio
		8	0.27	0.08	1.22	2.16	0.39	1.00
5		10	0.17	0.08	0.30	0.45	0.22	0.13
		16	1.34	0.08	1.80	8.42	0.44	0.96
		8	191.33	0.19	3.46	499.89	7.36	2.56
10	50	10	504.30	0.20	2.87	647.99	6.56	1.72
		16	254.36	0.20	2.37	396.86	5.18	2.00
		8	3612.72	0.49	2.70	5927.78	46.95	1.90
15		10	3576.25	0.36	3.64	5723.06	11.96	1.90
		16	6598.39	1.76	2.50	21037.11	7.77	2.00
		8	0.98	0.14	1.84	14.31	2.06	0.79
5		10	0.58	0.13	1.31	3.83	1.02	0.71
		16	2.95	0.14	0.97	14.98	1.31	0.50
		8	167.66	0.40	2.25	1297.92	31.00	1.20
10	100	10	44.33	0.36	2.43	193.03	18.23	1.70
		16	148.41	0.42	1.95	996.99	17.23	1.31
		8	816.91	3.93	2.00	3385.83	46.49	1.21
15		10	$>$ 12h	4.06	2.85	$>$ 12h	134.62	1.90
		16	8123.58	0.87	3.20	15802.72	72.47	2.18

Tabela 3.3: Tempo de *CPU* (seg) e desvio (%) - 20 localizações.

			$PLMC_3$			$PLMC_4$		
T	#J	#I^c	$PLIM$	Rl	% desvio	$PLIM$	Rl	% desvio
		8	0.14	0.06	1.22	0.58	0.20	1.00
5		10	0.13	0.06	0.30	0.34	0.19	0.13
		16	1.02	0.06	1.80	5.73	0.30	0.96
		8	46.25	0.11	3.46	149.31	1.54	2.56
10	50	10	153.55	0.16	2.87	334.00	2.00	1.72
		16	46.72	0.14	2.37	187.72	1.09	2.00
		8	1202.34	0.36	2.70	3993.95	3.14	1.90
15		10	848.45	0.33	3.64	969.36	4.04	1.90
		16	2808.12	0.64	2.50	11891.39	2.75	2.00
		8	0.89	0.11	1.84	5.27	1.43	0.79
5		10	0.28	0.11	0.31	0.66	0.38	0.81
		16	2.12	0.13	0.97	10.13	0.81	0.50
		8	73.53	0.36	2.25	231.91	5.10	1.20
10	100	10	33.69	0.38	2.43	322.91	4.38	1.70
		16	150.67	0.27	1.95	482.75	3.53	1.31
		8	382.81	0.68	2.00	485.27	18.85	1.21
15		10	$>$ 12h	1.58	2.85	$>$ 12h	16.03	1.90
		16	4646.67	0.83	3.20	10280.14	9.85	2.18

Tabela 3.4: Tempo de *CPU* (seg) e desvio (%) - 20 localizações.

			$PLMC_1$			$PLMC_2$		
T	$\#J$	$\#I^c$	$PLIM$	Rl	%desvio	$PLIM$	Rl	% desvio
		20	2.11	0.27	0.51	30.92	10.66	0.23
5		25	3.06	0.30	0.51	23.11	7.27	0.42
		45	50.02	0.32	0.60	204.00	8.68	0.27
		20	$>$ 12h	1.12	1.40	$>$ 12h	?0.13	1.00
10	100	25	$>$ 12h	1.09	1.20	$>$ 12h	6.20	0.85
		45	2883.53	0.92	0.71	2118.36	32.11	0.54
		20	$>$ 12h	2.88	0.92	$>$ 12h	624.25	0.57
15		25	$>$ 12h	3.02	0.97	$>$ 12h	257.45	0.79
		45	$>$ 12h	3.11	0.78	$>$ 12h	112.89	0.57

Tabela 3.5: Tempo de *CPU* (seg) e desvio (%) - 50 localizações.

			$PLMC_3$			$PLMC_4$		
T	$\#J$	$\#I^c$	$PLIM$	Rl	% desvio	$PLIM$	Rl	% desvio
		20	0.83	0.17	0.51	11.83	2.00	0.23
5		25	2.11	0.20	0.51	18.52	2.20	0.42
		45	31.00	0.23	0.60	279.25	3.14	0.27
		20	16516.25	3.01	1.40	$>$ 12h	19.53	1.00
10	100	25	$>$ 12h	2.72	1.20	$>$ 12h	14.61	0.85
		45	272.06	0.83	0.71	2383.62	13.22	0.54
		20	$>$ 12h	3.50	0.92	$>$ 12h	54.21	0.57
15		25	$>$ 12h	5.57	0.97	$>$ 12h	80.36	0.79
		45	$>$ 12h	1.74	0.78	$>$ 12h	29.61	0.57

Tabela 3.6: Tempo de *CPU* (seg) e desvio (%) - 50 localizações.

Analisando os resultados obtidos, verifica-se que o modelo $PLMC_1$ apresenta quase sempre, tempos de CPU melhores que $PLMC_2$ sendo a diferença, por vezes, significativa. Note-se que $PLMC_2$ tem mais $\#I \times \#J \times T$ restrições que $PLMC_1$, o que pode de alguma forma justificar a acentuada diferença de tempos, principalmente quando estão em causa instâncias de dimensão considerável. Por exemplo, nas maiores instâncias testadas (quando se considera $\#I$=50, $\#J$=100 e T=15), o modelo $PLMC_2$ tem mais $50 \times 100 \times 15 = 75\ 000$ restrições que o modelo $PLMC_1$.

No que diz respeito aos limites inferiores obtidos por relaxação linear, $PLMC_2$ é claramente melhor que $PLMC_1$. Estes resultados evidenciam a utilidade das restrições (3.14) e (3.15) no reforço da relaxação linear de $PLMC_1$.

Relativamente ao conjunto de restrições (3.16), verificou-se que a sua inclusão em cada um dos modelos $PLMC_1$ e $PLMC_2$, não melhorou o valor da relaxação linear. No entanto verificou-se um decréscimo substancial nos tempos de CPU quer do modelo inteiro misto quer da relaxação linear.

Façamos uma breve síntese para as instâncias com 10 localizações. A tabela 3.7 apresenta a média dos resultados para essas instâncias.

	PLIM	*Rl*	% desvio
$PLMC_1$	114.66	0.12	4.45
$PLMC_2$	235.10	4.84	3.02
$PLMC_3$	59.20	0.09	4.45
$PLMC_4$	61.15	1.02	3.02

Tabela 3.7: Média dos resultados - 10 localizações.

De acordo com a tabela 3.7, é evidente a diferença de tempo entre os modelos $PLMC_1$ e $PLMC_2$. Para este grupo de instâncias, $PLMC_2$ consumiu, em média, mais 105% do tempo consumido por $PLMC_1$, facto já justificado pela presença, em $PLMC_2$, de um número mais elevado restrições. Em relação à relaxação em programação linear, $PLMC_1$ consome, em média, cerca de 2.4% do tempo consumido por $PLMC_2$.

Repare-se no que acontece quando se adiciona, a cada um dos modelos, o conjunto de restrições (3.16). A redução de tempo nos dois modelos é muito significativa, sendo ainda mais acentuada para o modelo $PLMC_4$ que obtém a solução óptima em cerca de 26% do tempo conseguido por $PLMC_2$. Em relação ao modelo $PLMC_3$, gastou cerca de 51% do tempo consumido por $PLMC_1$. É de realçar que, em média, o modelo $PLMC_4$ obtém a solução óptima em apenas mais 2 segundos que $PLMC_3$. Desta forma, se pensarmos que o modelo $PLMC_4$ tem mais $\#I \times \#J \times T$ desigualdades, que se traduz em mais 15 000 restrições para a maior instância deste grupo, conclui-se que para este grupo de instâncias, o modelo $PLMC_4$ supera qualquer dos outros modelos, porque além da diferença do tempo médio ser quase insignificante relativamente ao modelo $PLMC_3$, apresenta melhores limites inferiores, com uma percentagem média de desvio da ordem dos 3 % contra os cerca de 4.5 % produzidos pelos modelos $PLMC_1$ e $PLMC_3$. A introdução das restrições (3.16) teve portanto, um impacto muito significativo no que se refere aos tempos de CPU.

As instâncias que mais contribuiram para os tempos produzidos por cada um dos modelos foram, naturalmente, as que consideram um horizonte de planeamento particionado em 15 períodos de tempo, por serem as que contêm um maior número de variáveis inteiras. De entre estas, realçam-se as que consideram $\#I^c = 5$ e $\#I^c = 8$. Em particular, as instâncias com 20 clientes, revelaram-se especialmente difíceis quando $\#I^c = 5$.

Vejamos o resumo para as instâncias com 20 localizações. Neste caso, uma vez que existe uma instância para a qual não se tem o tempo exacto e dado que é comum a todos os modelos, não se terá em conta a mesma no quadro da média dos resultados.

	$PLIM$	RL	% desvio
$PLMC_1$	1414.38	0.58	2.16
$PLMC_2$	3291.37	16.27	1.40
$PLMC_3$	611.61	0.28	2.16
$PLMC_4$	1726.73	3.50	1.40

Tabela 3.8: Média dos resultados - 20 localizações.

As instâncias com 20 localizações e 5 períodos de tempo foram resolvidas muito rapidamente. Para T=10 e T=15, verifica-se um aumento muito acentuado nos tempos de CPU especialmente quando se utilizam os modelos $PLMC_1$ e $PLMC_2$.

Observando a tabela 3.8 verifica-se que $PLMC_1$ requer, em média, cerca de 43 % do tempo requerido por $PLMC_2$. $PLMC_3$ requer, em média, cerca de 35 % do tempo requerido por $PLMC_4$. Note-se que já não existe a proximidade entre os tempos destes dois últimos modelos como se verificou nas instâncias com 10 localizações. Em relação à relaxação linear e no que diz respeito aos tempos de CPU, o aumento da dimensão das instâncias teve um impacto maior nos dois modelos considerados 'mais fortes', $PLMC_2$ e $PLMC_4$. $PLMC_3$ consumiu, em média, apenas 48 % do tempo consumido por $PLMC_1$, tendo $PLMC_4$ gasto, em média, cerca de 21 % do tempo gasto por $PLMC_2$ evidenciando, mais uma vez, a importante contribuição do conjunto de restrição (3.16). De realçar, ainda, a diminuição da média da percentagem dos desvios.

Também neste conjunto de instâncias, as dificuldades começam a surgir com o aumento do número de períodos que constituem o horizonte de planeamento, sendo particularmente críticas as situações que consideram $\#I^c \geq \frac{\#I}{2}$, em especial para T=15. É de realçar, ainda, o facto de, já neste conjunto, existir uma instância para a qual não foi possível encontrar a solução em tempo inferior a 12 horas por qualquer dos 4 modelos.

Tendo em conta os resultados obtidos para as instâncias com 50 localizações, de acordo com as tabelas 3.5 e 3.6, apenas para as instâncias que consideram um horizonte de planeamento dividido em 5 períodos de tempo se conseguiu uma solução em tempo inferior a 12 horas. Nem mesmo $PLMC_3$, que se tem revelado mais expedita, conseguiu melhor resultado que as restantes formulações, com excepção da instância com 10 períodos de tempo e 20 serviços existentes no início do horizonte de planeamento. Contudo, estes resultados não são de surpreender. Um aumento do número de localizações de 20 para 50, implica um aumento significativo do número de variáveis inteiras. O número máximo de variáveis inteiras no grupo de instâncias de 20 localizações é de 300, enquanto no grupo de instâncias com 50 localizações se atingem as 750.

3.3 Procedimento heurístico

3.3.1 Descrição e formalização da heurística

Numa tentativa de melhorar os tempos de execução apresentados nas tabelas atrás, nomeadamente para as instâncias de maior dimensão, apresenta-se de seguida uma heurística para obtenção de uma solução admissível para o problema e, consequentemente, um limite superior no valor óptimo do problema, o qual pode então ser fornecido como *input* para o procedimento de *Branch-and-Bound* do *general solver IlogCplex 9.0.* A heurística que vai ser apresentada revelar-se-á também útil no próximo capítulo.

Em termos gerais, a heurística é um procedimento iterativo que se inicia com a solução admissível em que se consideram todos os serviços em funcionamento, tanto quanto possível, ou seja, com os serviços do conjunto I^c a funcionar em todos os períodos do horizonte de planeamento e os serviços do conjunto I^o a funcionar desde o segundo período até ao fim do horizonte de planeamento. O objectivo de cada iteração é encontrar o serviço que, depois de lhe serem retirados períodos de funcionamento (desde que a solução resultante se mantenha admissível), conduz ao maior decréscimo relativamente ao valor da solução corrente. Na primeira iteração a solução corrente é a obtida na fase de inicialização.

Com base na solução corrente, procede-se à avaliação individual de cada um dos serviços, em termos da possibilidade de lhe retirar períodos de funcionamento (desde que a configuração resultante se mantenha admissível). Para cada um dos serviços $i \in I^c$, tenta-se, período a período, encerrar o mais cedo possível; para cada serviço $i \in I^o$, tenta-se abri-lo o mais tarde possível. A partir do momento em que, ao retirar um período de funcionamento, se obtenha uma solução não admissível, o serviço deixa de ser avaliado e passa-se à avaliação do serviço seguinte (voltando a ter como base a solução corrente).

Depois de avaliados todos os serviços, considera-se o par *serviço/novo período para encerramento ou abertura* que conduz à maior diminuição no valor da solução e altera-se a solução em conformidade. Passamos a ter uma nova solução corrente.

Na iteração seguinte, com base na nova solução corrente, procede-se à avaliação de cada um dos serviços ainda não fixados e, no final, fixa-se novamente o par *serviço/novo período de encerramento ou abertura* que, em relação ao valor da nova solução corrente, produz o maior decréscimo.

Prossegue-se assim, sucessivamente, até todos os serviços estarem fixados ou até que, depois de avaliados os serviços ainda não fixados, não se registe melhoria relativamente ao valor da solução corrente.

No final, a última solução corrente produzida é uma solução admissível para o problema, constituindo o seu valor um limite superior para o valor óptimo do problema em estudo.

De forma a facilitar a formalização da heurística em pseudo-linguagem, vamos definir os seguintes vectores:

$maxt[i]$ = último período de funcionamento do serviço $i \in I^c$ no horizonte de planeamento

$mint[i]$ = primeiro período de funcionamento do serviço $i \in I^o$ no horizonte de planeamento

$maxt[i] = T$, indicará que o serviço $i \in I^c$ está em funcionamento durante todo o horizonte de planeamento. O vector *maxt* tem dimensão igual ao número de serviços existentes no início do horizonte de planeamento.

$mint[i] \geq 2$ pelo facto de os novos serviços só poderem iniciar a sua actividade a partir do segundo período do horizonte de planeamento (inclusivé).

$mint[i] = T + 1$, significará que no local i não foi instalado qualquer serviço durante todo o horizonte de planeamento. O vector *mint* tem dimensão igual ao número de elementos do conjunto I^o.

Estes vectores são úteis na medida em que facilitam a representação de uma configuração dos serviços, nomeadamente de um ponto de vista computacional.

Já tínhamos visto como representar o conjunto dos serviços em actividade, num dado período, usando as variáveis z_{it} (conjunto O_t descrito na página 17). Vejamos agora como representá-lo através dos vectores *maxt* e *mint*. Um serviço $i \in I$, está em funcionamento no período $t \in TP$, se e só se, pertencer ao seguinte conjunto:

$$O_t = \{i \in I^c : maxt[i] \geq t\} \cup \{i \in I^o : mint[i] \leq t\}$$

O exemplo seguinte mostra como representar uma configuração para os serviços utilizando estes vectores.

Exemplo 3.1 *Considere-se uma instância do problema com 4 localizações tal que nas localizações 1 e 2 já existem serviços em funcionamento e as localizações 3 e 4 são candidatas à instalação de um novo serviço. Considere-se, ainda, um horizonte temporal particionado em 3 períodos de tempo.*

Tem-se então,

$I^c = \{1, 2\}$, $I^o = \{3, 4\}$, $TP = \{1, 2, 3\}$

Uma solução possível é manter o serviço 1 em funcionamento ao longo de todo o horizonte de planeamento e o serviço 2 até ao final do período 2. Quanto aos serviços 3 e 4, podem por exemplo, *iniciar a sua actividade nos períodos 2 e 3, respectivamente. Tudo isto se pode traduzir facilmente usando os vectores maxt e mint da forma que se segue:*

$$
\begin{array}{cc|ccc|ccc}
 & & \multicolumn{3}{c}{\textit{Períodos}} & & & \\
 & & \textit{1} & \textit{2} & \textit{3} & & & \\
\hline
 & \textit{1} & \blacksquare & \blacksquare & \blacksquare & \longrightarrow & & maxt[1] = 3 \\
\textit{Serviços} & \textit{2} & \blacksquare & \blacksquare & \square & \longrightarrow & & maxt[2] = 2 \\
 & \textit{3} & \square & \blacksquare & \blacksquare & \longrightarrow & & mint[3] = 2 \\
 & \textit{4} & \square & \square & \blacksquare & \longrightarrow & & mint[4] = 3 \\
\hline
\end{array}
$$

$\diamondsuit$

Antes formalizar a heurística em pseudo-linguagem, é necessário introduzir mais alguma notação:

CS = Custo da solução corrente.

$C(S)$ = Custo da solução resultante da alteração do estado de um serviço.

$SF[i] = 1$ se o serviço $i \in I$ está fixado e 0 caso contrário.

$$
ictrue = \begin{cases} 1, & \text{se o serviço que conduziu ao maior decréscimo numa dada iteração é um serviço de } I^c \\ 0, & \text{caso contrário} \end{cases}
$$

$$
iotrue = \begin{cases} 1, & \text{se o serviço que conduziu ao maior decréscimo numa dada iteração é um serviço de } I^o \\ 0, & \text{caso contrário} \end{cases}
$$

Naturalmente, ter-se-á sempre $ictrue + iotrue \leq 1$.

$servico$ = variável que identifica o serviço cuja alteração do estado produziu o maior decréscimo.

$periodo$ = variável que identifica o último (primeiro) período de funcionamento da variável $servico$.

Os custos CS e $C(S)$ determinam-se resolvendo, para cada período, um problema de transportes cujo valor é adicionado ao custo associado aos serviços em actividade. O problema de transportes é resolvido tendo em conta o número de serviços em actividade (os quais constituem as origens), a procura de cada cliente (que constituem os destinos) e os custos unitários de satisfação da procura que são os custos unitários de transporte. Os custos associados aos serviços podem ser obtidos através da matriz F_{it} (página 16).

No procedimento heurístico que estamos a apresentar e a que corresponde o algoritmo 3.3, sempre que há alteração do período em que um serviço é encerrado/instalado, é necessário averiguar se a solução resultante é ou não admissível para o problema. Para isso, basta verificar a admissibilidade da respectiva configuração de serviços. Esta verificação pode ser feita usando o procedimento 3.2, que devolve a variável *admissivel* com o valor 1 se a configuração for admissível e valor 0 caso contrário. Este procedimento é em tudo semelhante ao apresentado na página 15. Este último, no fundo, resulta de aplicar o algoritmo 3.2 à configuração em que todos os serviços estão em actividade tanto quanto possível.

Para cada período do horizonte de planeamento e de acordo com o conjunto O_t (página 29), verifica-se quais os serviços que estão em funcionamento e regista-se a capacidade total na variável *OfertaTot*. O mesmo acontecendo com a procura total que será guardada na variável *ProcTot*. Se em cada um dos períodos, a capacidade total é suficiente para satisfazer a procura total, a configuração é admissível, caso contrário, é não admissível.

Algorithm 3.2 Teste de admissibilidade de uma dada configuração

$admissivel = 1$
for $t \in TP$ **do**
  $ProcTot = 0$
  for $j \in J$ **do**
    $ProcTot = ProcTot + d_{jt}$
  $OfertaTot = 0$
  for $i \in I^c$ **do**
    if $maxt[i] \geq t$ **then**
      $OfertaTot = OfertaTot + Q_i$
  for $i \in I^o$ **do**
    if $mint[i] \leq t$ **then**
      $OfertaTot = OfertaTot + Q_i$
  if $ProcTot > OfertaTot$ **then**
    $admissivel = 0$. Stop.

Observação *Embora na apresentação do algoritmo que se segue esteja implícita uma ordem pela qual é feita a avaliação dos serviços, deve referir-se que a ordem é indiferente, conduzindo à mesma solução admissível.* ♣

Observação *A procura total em cada período resulta dos dados de cada instância e não é modificável pelo procedimento. Assim, logo no início, e para evitar a repetição de cálculos desnecessários, determina-se a procura total em cada período sendo os valores obtidos guardados num vector de dimensão T.* ♣

Algorithm 3.3 Procedimento para obtenção de uma solução admissível

for $i \in I^c$ **do**
  $maxt[i] = T$
for $i \in I^o$ **do**
  $mint[i] = 2$
calcular CS
for $i \in I$ **do**
  $\mathbf{SF}[i] = 0$
while $h > 0$ **do**
  h=0, ictrue=0, iotrue=0
  for $i \in I^c$ e $\mathbf{SF}[i] = 0$ **do**
    $t = maxt[i]$
    while $maxt[i] > 1$ **do**
      $maxt[i] = maxt[i] - 1$
      Aplicar o algoritmo 3.2
      if $admissivel = 1 \wedge CS - C(S) > h$ **then**
        $h = CS - C(S)$
        $servico = i$
        $periodo = maxt[i]$
        $ictrue = 1$
      else
        Saír do ciclo while
    $maxt[i] = t$
  for $i \in I^o \wedge \mathbf{SF}[i] = 0$ **do**
    $t = mint[i]$
    while $mint[i] \leq T$ **do**
      $mint[i] = mint[i] + 1$
      Aplicar o algoritmo 3.2
      if $admissivel = 1$ and $CS - C(S) > h$ **then**
        $h = CS - C(S)$
        $servico = i$
        $periodo = mint[i]$
        $iotrue = 1$
        $ictrue = 0$
      else
        Saír do ciclo while
    $mint[i] = t$
  if $ictrue = 1$ **then**
    $maxt[servico] = periodo$
    $CS = CS - h$
    $\mathbf{SF}[servico] = 1$
  else
    if $iotrue = 1$ **then**
      $mint[servico] = periodo$
      $CS = CS - h$
      $\mathbf{SF}[servico] = 1$

Seguidamente vamos ver um exemplo de aplicação da heurística a uma pequena instância do problema de *phase-in/phase-out* em estudo. Refira-se que os problemas de transportes que surgem ao longo do exemplo, foram resolvidos utilizando o algoritmo *primal-dual* usado por Saldanha da Gama [46] que implementou em linguagem de programação *C* o algoritmo descrito em Syslo *et al.* [48] implementado originalmente em linguagem *Pascal*.

Exemplo 3.2 *Considere-se uma instância com 3 localizações. Em cada um dos locais 1 e 2, já existe um serviço em actividade sendo o local 3 candidato a uma possível instalação. 5 clientes precisam ver satisfeita a sua procura nos 3 períodos que dividem o horizonte de planeamento. Tem-se então,*

$I^c = \{1, 2\}$, $I^o = \{3\}$, $I = \{1, 2, 3\}$, $TP = \{1, 2, 3\}$

Consideremos para cada um dos serviços uma capacidade limitada dada por:

i	1	2	3
Q_i	107	237	190

Consideremos como custos de operação, abertura e encerramento os seguintes valores:

f_{it}	1	2	3
1	938	991	1066
2	2221	2275	2490
3	-	2134	2190

g_{it}	1	2	3
1	1630	1684	-
2	5584	5725	-

h_{it}	1	2	3
3	-	1524	1618

Suponhamos que a procura dos clientes em cada período é dada por:

d_{jt}	1	2	3
1	19	20	18
2	79	82	77
3	81	88	90
4	29	30	29
5	18	17	18

Vamos assumir, ainda, que cada uma das matrizes seguintes representa o custo de satisfação total da procura em cada um dos períodos por cada um dos serviços.

c_{ij1}	1	2	3	4	5
1	185	179	683	214	102
2	287	568	1203	45	62

c_{ij2}	1	2	3	4	5
1	197	168	624	229	110
2	302	596	1299	49	58
3	178	471	625	342	159

c_{ij3}	1	2	3	4	5
1	182	160	633	247	104
2	273	545	1322	45	62
3	159	442	634	335	172

Inicialmente, começa-se por construir a solução admissível em que todos os serviços se encontram em funcionamento tanto quanto possível ao longo do horizonte de planeamento. Esta é a primeira solução corrente. Esta solução é admissível desde que a instância tenha pelo menos uma solução admissível. No caso presente é fácil verificar que se todos os serviços estiverem sempre que possível em actividade, a capacidade total é suficiente para satisfazer a procura total em todos os períodos.

	1	2	3
1	■	■	■
2	■	■	■
3	□	■	■

Para determinar o custo da solução associada a esta configuração, basta adicionar aos custos associados aos serviços, os custos de satisfação da procura. Os primeiros são obtidos facilmente através da matriz F_{it}. Para obter os segundos é necessário resolver um problema de transportes para cada período do horizonte de planeamento, tendo em atenção que as origens são constituídas pelos serviços em funcionamento nesse período, e os destinos pelos clientes, os quais têm associada a procura no período em causa.

De acordo com (3.1), (3.2) e (3.3) (página 16), pode construir-se a matriz F_{it}:

F_{it}	1	2	3
1	$g_{11} + f_{11}$	$g_{12} + f_{11} + f_{12}$	$f_{11} + f_{12} + f_{13}$
2	$g_{21} + f_{21}$	$g_{22} + f_{21} + f_{22}$	$f_{21} + f_{22} + f_{23}$
3	-	$h_{32} + f_{32} + f_{33}$	$h_{33} + f_{33}$

=

F_{it}	1	2	3
1	2568	3613	2995
2	7805	10221	6986
3	-	5848	3808

Os serviços 1 e 2 estão em funcionamento até ao final do período 3, pelo que tem que se considerar F_{13} + F_{23}. O serviço 3 inicia o seu funcionamento no segundo período, pelo que o custo correspondente é F_{32}. Designando por CC o custo associado à configuração obtém-se

$$CC = F_{13} + F_{23} + F_{32} = 2\,995 + 6\,986 + 5\,848 = 15\,829$$

A este valor acrescenta-se o valor óptimo de cada um dos problemas de transporte referentes a cada período.

período 1:

i	1	2
Q_i	107	237

j	1	2	3	4	5
d_{j1}	19	79	81	29	18

As matrizes de custos que constituem os dados da instância em análise, representam o custo total de satisfazer, num dado período, toda procura de um dado cliente por um dado serviço. Em termos dos problemas de transportes o importante é conhecer os valores dos custos unitários de afectação dos clientes aos serviços em funcionamento. Para o primeiro período, tendo em conta que os serviços em funcionamento são os serviços 1 e 2, ter-se-á

C_{ij1}	1	2	3	4	5
1	$\frac{185}{19}$	$\frac{179}{79}$	$\frac{683}{81}$	$\frac{214}{29}$	$\frac{102}{18}$
2	$\frac{287}{19}$	$\frac{568}{79}$	$\frac{1203}{81}$	$\frac{45}{29}$	$\frac{62}{18}$

A solução para o problema de transportes associado ao primeiro período é:

	cliente 1	cliente 2	cliente 3	cliente 4	cliente 5
serviço 1	19	7	81	0	0
serviço 2	0	72	0	29	18

com custo

$$C_1 = (19 \times \tfrac{185}{19}) + (7 \times \tfrac{179}{79}) + (81 \times \tfrac{683}{81}) + (72 \times \tfrac{568}{79}) + (29 \times \tfrac{45}{29}) + (18 \times \tfrac{62}{18}) = 1\ 508.5$$

período 2:

No período 2, de acordo com a solução, todos os serviços estão em funcionamento. Desta forma ter-se-á um problema de transportes com 3 origens e 5 destinos, definidos por:

i	1	2	3
Q_i	107	237	190

j	1	2	3	4	5
d_{j2}	20	82	88	30	17

C_{ij2}	1	2	3	4	5
1	$\frac{197}{20}$	$\frac{168}{82}$	$\frac{624}{88}$	$\frac{229}{30}$	$\frac{110}{17}$
2	$\frac{302}{20}$	$\frac{596}{82}$	$\frac{1299}{88}$	$\frac{49}{30}$	$\frac{58}{17}$
3	$\frac{178}{20}$	$\frac{471}{82}$	$\frac{625}{88}$	$\frac{342}{30}$	$\frac{159}{17}$

A solução deste problema é a seguinte:

	cliente 1	cliente 2	cliente 3	cliente 4	cliente 5
serviço 1	0	82	25	0	0
serviço 2	0	0	0	30	17
serviço 3	20	0	63	0	0

$$C_2 = (82 \times \tfrac{168}{82}) + (25 \times \tfrac{624}{88}) + (30 \times \tfrac{49}{30}) + (17 \times \tfrac{58}{17}) + (20 \times \tfrac{178}{20}) + (63 \times \tfrac{625}{88}) = 1\ 077.7$$

período 3:

No período 3 todos os serviços estão disponíveis. O problema de transportes é definido por:

i	1	2	3
Q_i	107	237	190

j	1	2	3	4	5
d_{j3}	18	77	90	29	18

C_{ij3}	1	2	3	4	5
1	$\frac{182}{18}$	$\frac{160}{77}$	$\frac{633}{90}$	$\frac{247}{29}$	$\frac{104}{18}$
2	$\frac{273}{18}$	$\frac{545}{77}$	$\frac{1322}{90}$	$\frac{45}{29}$	$\frac{62}{18}$
3	$\frac{159}{18}$	$\frac{442}{77}$	$\frac{634}{90}$	$\frac{335}{29}$	$\frac{172}{18}$

sendo a sua solução:

	cliente 1	*cliente 2*	*cliente 3*	*cliente 4*	*cliente 5*
serviço 1	*0*	*77*	*30*	*0*	*0*
serviço 2	*0*	*0*	*0*	*29*	*18*
serviço 3	*18*	*0*	*60*	*0*	*0*

$$C_3 = (77 \times \tfrac{160}{77}) + (30 \times \tfrac{633}{90}) + (29 \times \tfrac{45}{29}) + (18 \times \tfrac{62}{18}) + (18 \times \tfrac{159}{18}) + (60 \times \tfrac{634}{90}) = 1\ 059.6$$

O custo da solução inicial (que fica como solução corrente) é então,

custo = 15 829 + C1 + C2 + C3 = 19 474.8

$\boldsymbol{SF}[i] = 0, \forall i \in I$, *h = 0, ictrue=0, iotrue=0*

Começando por avaliar o primeiro serviço de I^c*:*

$i = 1$ *e* $\boldsymbol{SF}[1] = 0$

$t = maxt[1] = 3 > 1$, *então pode reduzir-se um período de funcionamento resultando na seguinte configuração:*

	1	*2*	*3*
1	■	■	□
2	■	■	■
3	□	■	■

Neste caso não há garantia de que a solução seja admissível. Há que verificar a admissibilidade. Para tal, basta verificar se no período que se retirou ao serviço, a soma das capacidades dos serviços em actividade é não inferior à procura total nesse período.

período 3:

$$Q_2 + Q_3 = 237 + 190 = 427 > \sum_{j=1}^{5} d_{j3} = 232 \checkmark$$

A configuração é admissível. O custo da solução associada é de 20 373.24, obtido, tal como para a solução anterior ou seja, adicionando os custos associados aos serviços, aos valores óptimos dos 3 problemas de transportes associados a cada um dos períodos considerando como origens os serviços em actividade: no período 1, os serviços 1 e 2; no período 2, os serviços 1, 2 e 3. Finalmente, no período 3, os serviços 3 e 4. O custo desta solução é superior ao custo da solução corrente, pelo que esta última é uma melhor solução. Como neste momento $maxt[1] = 2 > 1$ *pode reduzir-se para* $maxt[1] = 1$, *conduzindo à seguinte configuração:*

	1	2	3
1	■	□	□
2	■	■	■
3	□	■	■

A solução correspondente é admissível
custo = 19 631.53 > 19 474.8

Como $maxt[1] = 1 \not> 1$, *já não existe a possibilidade de retirar mais períodos de funcionamento ao serviço, pelo que o serviço 1 deixa de ser avaliado. Passa-se ao serviço seguinte partindo da solução corrente, ou seja, voltando a pôr* $maxt[1] = t = 3$.

$i = 2$ *e* $\boldsymbol{SF}[2] = 0$

$t = maxt[2] = 3 > 1$. *Retira-se um período de funcionamento ao serviço.*

$maxt[2] = 2$*:*

	1	2	3
1	■	■	■
2	■	■	□
3	□	■	■

A solução correspondente é admissível
custo = 23 003.83 > 19 474.8

$maxt[2] = 2 > 1$. *Faz-se* $maxt[2] = 1$*:*

	1	2	3
1	■	■	■
2	■	□	□
3	□	■	■

A solução correspondente é admissível
custo = 20 887.59 > 19 474.8

Não se pode encerrar mais cedo o serviço pois $maxt[2] = 1 \not> 1$. *Repõe-se a solução corrente fazendo* $maxt[2] = t = 3$.
Passa-se agora ao único serviço do conjunto I^o.

$i = 3$ *e* $\boldsymbol{SF}[3] = 0$

$t = mint[3] = 2$

$mint[3] = 2 \leq 3$, *então faz-se* $mint[3] = 3$ *que produz a seguinte configuração:*

	1	2	3
1	■	■	■
2	■	■	■
3	□	□	■

A solução correspondente é admissível
custo = 17 884.44 < 19 474.8 sendo a diferença de 1 590.36 > h
então fixa-se h= 1 590.36, ictrue=0, iotrue=1

Como $mint[3] = 3 \leq 3$, *pode retirar-se um período de funcionamento ao serviço fazendo* $mint[3] = 4$*:*

	1	2	3
1	■	■	■
2	■	■	■
3	□	□	□

A solução correspondente é admissível
custo = 14 490.75 < 19 474.8 sendo a diferença de 4 984.05 > h
então fixa-se h= 4 984.05

$mint[3] = 4 > 3$ *e todos os serviços foram avaliados.*

O serviço 3 foi o que originou o maior decréscimo pelo que será o serviço a ser fixado em $\boldsymbol{SF}$*.*

$\boldsymbol{SF}[3] = 1$

Como h > 0, volta-se a analisar todos os serviços i para os quais $\boldsymbol{SF}[i] = 0$ *tendo por base a nova solução corrente.*

A nova solução corrente é:

	1	2	3
1	■	■	■
2	■	■	■
3	□	□	□

custo corrente = CS= 14 490.75

h = 0

$i = 1$ *e* $\boldsymbol{SF}[1] = 0$

$t = maxt[1] = 3 > 1$*, então pode-se reduzir um período de funcionamento ao serviço 1, resultando na seguinte configuração:*

	1	2	3
1	■	■	□
2	■	■	■
3	□	□	□

A solução correspondente é admissível
custo = 15 881.78 > 14 490.75

Neste momento $maxt[1] = 2 > 1$ *então reduz-se para* $maxt[1] = 1$*:*

	1	2	3
1	■	□	□
2	■	■	■
3	□	□	□

A solução correspondente é admissível
custo = 15 611.53 > 14 490.75

Como $maxt[1] = 1 \ngtr 1$*, repõe-se a solução corrente fazendo* $maxt[1] = t = 3$ *e passa-se à avaliação do serviço seguinte.*

$i = 2$ *e* $\boldsymbol{SF}[2] = 0$

$maxt[2] = 3 > 1$. Pode, então, reduzir-se para $maxt[2] = 2$, que conduz a uma configuração não admissível:

	1	2	3
1	■	■	■
2	■	■	□
3	□	□	□

A configuração é não admissível e qualquer outra obtida a partir desta removendo mais períodos de operação ao serviço 2 é não admissível.

Volta-se à solução corrente fazendo $maxt[2] = t = 3$ e passa-se à avaliação do serviço 3.

$i = 3$ e $\boldsymbol{SF}[3] = 1 \Longrightarrow$ não pode ser avaliado.

Como foram avaliados todos os serviços ainda não fixados e h=0, a melhor solução admissível encontrada é a actual solução corrente:

	1	2	3
1	■	■	■
2	■	■	■
3	□	□	□

Configuração admissível obtida

Custo associado aos serviços:

$CC = F_{13} + F_{23} + 0 = 2\ 995 + 6\ 986 = 9\ 981$

Dados do problema de transportes associado ao período 1:

i	1	2
Q_i	107	237

j	1	2	3	4	5
d_{j1}	19	79	81	29	18

C_{ij1}	1	2	3	4	5
1	$\frac{185}{19}$	$\frac{179}{79}$	$\frac{683}{81}$	$\frac{214}{29}$	$\frac{102}{18}$
2	$\frac{287}{19}$	$\frac{568}{79}$	$\frac{1203}{81}$	$\frac{45}{29}$	$\frac{62}{18}$

A solução para este problema é:

	cliente 1	cliente 2	cliente 3	cliente 4	cliente 5
serviço 1	19	7	81	0	0
serviço 2	0	72	0	29	18

com custo

$$C_1 = (19 \times \frac{185}{19}) + (7 \times \frac{179}{79}) + (81 \times \frac{683}{81}) + (72 \times \frac{568}{79}) + (29 \times \frac{45}{29}) + (18 \times \frac{62}{18}) = 1\ 508.5$$

No período 2, os serviços em funcionamento são os serviços 1 e 2. Os dados do problema de transportes associado a este período são:

i	1	2
Q_i	107	237

j	1	2	3	4	5
d_{j2}	20	82	88	30	17

C_{ij2}	1	2	3	4	5
1	$\frac{197}{20}$	$\frac{168}{82}$	$\frac{624}{88}$	$\frac{229}{30}$	$\frac{110}{17}$
2	$\frac{302}{20}$	$\frac{596}{82}$	$\frac{1299}{88}$	$\frac{49}{30}$	$\frac{58}{17}$

cuja solução é a seguinte:

	cliente 1	*cliente 2*	*cliente 3*	*cliente 4*	*cliente 5*
serviço 1	*19*	*0*	*88*	*0*	*0*
serviço 2	*1*	*82*	*0*	*30*	*17*

$$C_2 = (19 \times \tfrac{197}{20}) + (88 \times \tfrac{624}{88}) + (1 \times \tfrac{302}{20}) + (82 \times \tfrac{596}{82}) + (30 \times \tfrac{49}{30}) + (17 \times \tfrac{58}{17}) = 1\ 529.25$$

Também no período 3 são os serviços 1 e 2 que estão disponíveis. Os dados do problema de transportes associado a este período são:

i	*1*	*2*
Q_i	*107*	*237*

j	*1*	*2*	*3*	*4*	*5*
d_{j3}	*18*	*77*	*90*	*29*	*18*

C_{ij3}	*1*	*2*	*3*	*4*	*5*
1	$\frac{182}{18}$	$\frac{160}{77}$	$\frac{633}{90}$	$\frac{247}{29}$	$\frac{104}{18}$
2	$\frac{273}{18}$	$\frac{545}{77}$	$\frac{1322}{90}$	$\frac{45}{29}$	$\frac{62}{18}$

sendo a solução:

	cliente 1	*cliente 2*	*cliente 3*	*cliente 4*	*cliente 5*
serviço 1	*17*	*0*	*90*	*0*	*0*
serviço 2	*1*	*77*	*0*	*29*	*18*

$$C_3 = (17 \times \tfrac{182}{18}) + (90 \times \tfrac{633}{90}) + (1 \times \tfrac{273}{18}) + (77 \times \tfrac{545}{77}) + (29 \times \tfrac{45}{29}) + (18 \times \tfrac{62}{18}) = 1\ 472$$

O custo total da solução é:

$custo = CC + C1 + C2 + C3 = 14\ 490.75$

Determinadas as soluções dos problemas de transporte pode, finalmente, apresentar-se a solução admissível em termos das variáveis do problema, $\boldsymbol{z}$ e $\boldsymbol{x}$.

Solução:

$valor = 14\ 490.75$

$z_{it} = 0,\ i = 1,\ 2,\ 3\ ;\ t = 1,\ 2,\ 3$

$x_{111} = x_{131} = x_{241} = x_{251} = x_{132} = x_{222} = x_{242} = x_{252} = x_{133} = x_{223} = x_{243} = x_{253} = 1$

$x_{121} = \frac{7}{79};\ x_{221} = \frac{72}{79};\ x_{112} = \frac{19}{20};\ x_{212} = \frac{1}{20};\ x_{113} = \frac{17}{18};\ x_{213} = \frac{1}{18}$

$x_{ijt} = 0$, *para os restantes.* $\diamondsuit$

3.3.2 Experiência computacional

Nesta secção, apresentam-se os resultados de alguma experimentação efectuada para procurar avaliar a qualidade da heurística proposta. Concretamente, apresenta-se o tempo de *CPU* para obtenção de uma solução admissível para cada uma das instâncias usadas nos testes computacionais cujos resultados se apresentaram na secção 3.2. Dado que é conhecido o valor óptimo de cada uma das instâncias, apresenta-se ainda o desvio relativo (em percentagem) do limite superior associado à solução admissível dada pela heurística. Este desvio define-se, para cada instância por:

$$\frac{V_H - vOpt}{vOpt} \times 100$$

sendo V_H o valor da solução admissível obtida e $vOpt$ o valor óptimo da instância em causa.

A tabela 3.9 apresenta os valores correspondentes às instâncias com 10 localizações. Para cada instância apresenta-se o tempo de *CPU* (em segundos) e o desvio relativo do limite superior obtido.

T	$\#J$	$\#I^c$	tempo (seg)	% desvio
		3	0.03	0.00
5		5	0.06	0.00
		8	0.06	3.10
		3	0.14	5.90
10	20	5	0.19	3.80
		8	0.20	0.00
		3	0.53	0.65
15		5	0.28	2.50
		8	3.33	8.40
		3	0.83	0.00
5		5	1.01	0.26
		8	1.25	0.80
		3	3.75	14.00
10	50	5	3.64	0.00
		8	3.80	8.50
		3	6.36	4.68
15		5	12.23	3.80
		8	11.66	3.00
		3	2.48	0.75
5		5	2.75	0.00
		8	2.89	0.00
		3	10.00	7.10
10	100	5	8.05	1.36
		8	9.08	7.83
		3	25.83	5.40
15		5	21.22	5.60
		8	35.67	3.17

Tabela 3.9: *CPU* e desvio (%) - 10 localizações.

As tabelas 3.10 e 3.11 apresentam o tempo de *CPU* (em segundos) e os desvios relativos para as instâncias com 20 e 50 localizações, respectivamente.

T	$\#J$	$\#I^c$	tempo (seg)	% desvio
		8	9.53	0.00
5		10	7.20	0.00
		16	10.63	0.42
		8	36.75	6.70
10	50	10	48.59	6.50
		16	55.52	4.00
		8	150.16	7.10
15		10	77.08	2.10
		16	78.69	4.25
		8	30.75	0.61
5		10	26.31	3.10
		16	22.28	0.37
		8	108.81	3.97
10	100	10	139.55	2.50
		16	103.45	2.30
		8	318.76	12.50
15		10	241.30	6.90
		16	203.30	0.00

Tabela 3.10: *CPU* e desvio (%) - 20 localizações

T	$\#J$	$\#I^c$	tempo (seg)	% desvio
		20	83.86	2.50
5		25	92.92	4.22
		45	106.52	1.23
		20	404.09	6.50
10	100	25	371.09	4.40
		45	648.27	3.40
		20	667.69	7.63
15		25	1124.34	4.40
		45	1194.95	1.94

Tabela 3.11: *CPU* e desvio (%) - 50 localizações

Tendo em conta a tabela 3.9 (instâncias com 10 localizações) verifica-se que em cerca de 40 % dos casos, o desvio é inferior a 1 %, tendo-se obtido a solução óptima em 26 % das instâncias consideradas neste grupo. Com excepção da instância com 50 clientes e 10 períodos de tempo, nenhuma instância ultrapassou os 8.5 % e apenas 26 % ultrapassam os 5 %.

Quanto às instâncias do grupo com 20 localizações (tabela 3.10), pode dizer-se que a heurística mantém uma *performance* satisfatória. Em $\frac{1}{3}$ das instâncias obtiveram-se desvios inferiores a 1 %, conseguindo-se em algumas delas obter a solução óptima. Excluindo a instância que obteve 12.5 % de desvio, nenhuma outra ultrapassou os 7.1 %. Quantos aos tempos, e apesar do aumento natural devido ao aumento da dimensão, podem considerar-se aceitáveis se se tiver em conta a dimensão já considerável destas instâncias.

Relativamente ao grupo de instâncias com 50 localizações, embora se verifique um aumento, de alguma forma substancial, nos tempos de *CPU* nomeadamente para as instâncias com 15 períodos, a percentagem do desvio mostra que a qualidade das soluções obtidas não parecem depender significativamente da dimensão das instâncias.

Em termos da heurística, a qualidade da solução admissível obtida e o tempo necessário para isso, não parecem estar directamente relacionados com o número de serviços existentes no início do horizonte de planeamento. Independentemente da instância, a solução obtida é, em geral, caracterizada da seguinte forma : os serviços de I^c ou estão a funcionar ao longo de todo o horizonte de planeamento ou só funcionam no primeiro período; os serviços de I^o ou começam a funcionar no segundo período ou não funcionam em qualquer período do horizonte de planeamento.

3.4 Uso de um *upper-cut-off*

Tal como se mencionou na introdução deste capítulo, o valor de cada uma das soluções admissíveis produzidas pela heurística foi usado como *upper-cut-off* inicial no procedimento de *Branch--and-Bound* do *IlogCplex 9.0* [13] de forma a tentar reduzir os tempos de *CPU* apresentados na secção 3.2, nomeadamente nos modelos com piores resultados. Assim, nesta secção apresentam-se os resultados obtidos para os modelos $PLMC_1$ e $PLMC_2$ usando o valor da heurística como *upper-cut-off* inicial.

A tabela 3.12 a seguir apresenta os resultados para as instâncias com 10 localizações; a tabela 3.13 apresenta o mesmo tipo de resultados mas para as instâncias com 20 localizações. De referir apenas que, aos tempos fornecidos pelo *IlogCplex 9.0*, foram acrescidos os tempos de execução da heurística.

			$PLMC_1$		$PLMC_2$	
T	$\#J$	$\#I^c$	$PLIM$	*Rl*	$PLIM$	*Rl*
		3	0.06	0.04	0.09	0.02
5		5	0.23	0.04	0.25	0.05
		8	0.20	0.04	0.33	0.05
		3	2.30	0.05	6.28	0.23
10	20	5	1.81	0.04	3.02	0.18
		8	0.94	0.03	1.69	0.13
		3	101.87	0.08	127.64	0.66
15		5	764.02	0.08	1137.59	0.64
		8	6.94	0.07	11.06	0.32
		3	0.92	0.05	1.23	0.14
5		5	1.47	0.05	2.73	0.11
		8	1.83	0.03	2.59	0.22
		3	34.48	0.11	30.17	0.92
10	50	5	5.98	0.08	12.88	0.83
		8	59.56	0.10	165.41	1.05
		3	198.16	0.14	521.22	6.80
15		5	228.70	0.16	289.33	4.79
		8	968.47	0.24	1555.94	2.95
		3	3.14	0.08	4.95	0.66
5		5	2.86	0.08	3.31	0.32
		8	3.58	0.06	4.52	0.41
		3	41.50	0.16	59.75	5.98
10	100	5	23.19	0.18	60.73	6.00
		8	11.02	0.18	21.66	3.26
		3	222.22	0.31	626.09	29.12
15		5	535.11	0.50	1636.45	29.93
		8	179.02	0.36	417.02	24.48

Tabela 3.12: *CPU* (seg) para instâncias com 10 localizações

Considerando as instâncias com 10 localizações e comparativamente com a tabela 3.1 (página

			$PLMC_1$		$PLMC_2$	
T	$\#J$	$\#I^c$	$PLIM$	Rl	$PLIM$	Rl
		8	9.77	0.05	11.59	0.26
5		10	7.36	0.08	7.59	0.16
		16	11.89	0.08	18.31	0.31
		8	228.05	0.18	540.91	7.35
10	50	10	549.17	0.19	712.50	3.51
		16	308.95	0.19	450.31	1.81
		8	3374.72	0.50	6077.06	46.93
15		10	3300.08	0.36	5585.36	8.83
		16	6248.91	0.60	16807.00	7.88
		8	31.75	0.14	44.88	1.90
5		10	26.86	0.13	30.02	0.91
		16	25.05	0.13	36.58	1.24
		8	275.56	0.40	1382.19	30.77
10	100	10	183.84	0.36	345.61	15.11
		16	251.62	0.50	1068.75	12.60
		8	1129.69	3.95	3706.72	43.30
15		10	> 12h	0.83	> 12h	127.36
		16	6584.94	0.85	12677.91	72.21

Tabela 3.13: *CPU* (seg) para instâncias com 20 localizações

22), pode constatar-se que a introdução de um *upper-cut-off* inicial não teve um impacto positivo, nem em $PLMC_1$, nem em $PLMC_2$. Os tempos de resolução do problema aumentaram em muitas instâncias e por vezes de forma significativa. Para $PLMC_1$, nas instâncias com 50 e 100 clientes e 15 períodos de tempo, verificou-se uma ligeira melhoria, mas nada de significativo. Relativamente a $PLMC_2$, em algumas instâncias verificaram-se pequenos decréscimos no tempo mas para outras verificou-se um aumento (embora ligeiro).

Com respeito à relaxação linear e no que se refere a $PLMC_1$, os tempos também aumentaram embora apenas algumas décimas de segundo no total. No modelo $PLMC_2$ sim, verificou-se um aumento de quase 11 segundos no total.

No grupo de instâncias com 20 localizações revelam-se algumas melhorias, nomeadamente nas instâncias com 15 períodos de tempo. No entanto, nas restantes verificam-se tempos superiores aos obtidos antes da utilização do *upper-cut-off*. Vejamos uma síntese com os valores totais para as instâncias com 10 e 20 localizações:

	$PLIM$ (seg)	Rl (seg)
$PLMC_1$	24 044.53	9.83
$PLMC_2$	55 953.33	276.64

Sem upper-cut-off

	$PLIM$ (seg)	Rl (seg)
$PLMC_1$	22 548.21	8.69
$PLMC_2$	49 503.29	255.08

Com upper-cut-off

De referir que não foi incluída a instância para a qual não se obteve a solução em tempo inferior a 12 horas ($\#I = 20/\ \#J = 100/\ T = 15/\ \#I^c = 10$). Com base nos quadros acima verifica-se uma redução no tempo total para $PLMC_1$ e $PLMC_2$, quer do problema inteiro quer da relaxação linear, sendo mais expressiva em $PLMC_2$. Mesmo assim não se obteve uma solução em tempo inferior a 12 horas para a instância acima referida.

Relativamente ao grupo de 50 localizações e à semelhança do que aconteceu com os modelos $PLMC_1$ e $PLMC_2$ na secção 3.2 antes da utilização de um *upper-cut-off* inicial, não foi conseguida, para a maioria das instâncias, uma solução em tempo inferior a 12 horas.

3.5 Conclusão

Neste capítulo foram apresentadas e analisadas quatro formulações equivalentes em programação linear inteira mista para o problema de *phase-in/phase-out* em estudo: $PLMC_1$, $PLMC_2$, $PLMC_3$ e $PLMC_4$. As formulações foram codificadas usando a linguagem de modelação *IlogConcert 2.0* [12] e avaliadas usando um conjunto de instâncias geradas aleatoriamente tal como descrito na página 20, utilizando o procedimento de *Branch-and-Bound* do *general solver IlogCplex 9.0* [13].

Através da experiência computacional pôde constatar-se que de entre os modelos $PLMC_1$ e $PLMC_2$ (recordando que o segundo resulta do primeiro adicionando as desigualdades (3.14) e (3.15) apresentadas na página 19), $PLMC_1$ apresenta uma melhor *performance* no que se refere a tempos de *CPU*, sendo, no entanto, preferível o segundo em termos de relaxação em programação linear, pois fornece melhores limites inferiores para o problema. A introdução, em cada um destes modelos, do conjunto de restrições (3.16) (o que deu origem aos modelos $PLMC_3$ e $PLMC_4$), não permitiu melhorar os limites inferiores já obtidos mas permitiu reduzir significativamente os tempos de *CPU*. O impacto foi mais significativo quando se passou de $PLMC_2$ para $PLMC_4$. Há, no entanto, que acentuar a incapacidade de o procedimento de *Branch-and-Bound* do *general solver* resolver em tempo inferior a 12 horas (valor máximo considerado) a maioria das instâncias com 50 localizações. Foi ainda apresentado um procedimento heurístico que permitiu obter um limite superior para o problema. Testes computacionais efectuados usando o mesmo conjunto de instâncias, permitiram concluir que a qualidade das soluções admissíveis obtidas é bastante razoável e mais ainda se se tiver em conta a estabilidade da percentagem do desvio relativamente ao valor óptimo mesmo quando consideradas instâncias de maior dimensão.

A última parte do capítulo apresenta os resultados obtidos quando se utilizou um *upper-cut-off* inicial nos modelos $PLMC_1$ e $PLMC_2$. Verificou-se que para $PLMC_2$, os tempos de *CPU* reduziram em algumas instâncias, nomeadamente nas que têm 20 localizações. Em outras instâncias o tempo de *CPU* aumentou, principalmente nas que têm 5 e 10 períodos de tempo. No total verificou-se, de facto, um decréscimo nos tempos de *CPU*. Relativamente a $PLMC_1$, não se verificaram melhorias nas instâncias com 10 localizações quando se considerou um *upper-cut-off* inicial. Nas instâncias com 20 localizações ocorreu uma ligeira diminuição mas quase imperceptível. O tempo de execução da heurística é, em geral, responsável pelos aumentos verificados.

Capítulo 4

Decomposição de Benders

4.1 Introdução

Neste capítulo é proposta outra abordagem para resolver o problema de phase-in/phase-out multi-periódico em estudo nesta dissertação. Concretamente, estudar-se-á a utilização do método da decomposição de Benders.

A aplicação de uma técnica de decomposição tem como objectivo principal reduzir a dimensão de um problema e, consequentemente, tentar resolvê-lo em tempo computacional inferior ao que à partida seria necessário para resolver o problema original. É com este propósito que se propõe a aplicação da técnica da decomposição de Benders. Embora esta técnica seja conhecida pelo enorme esforço computacional que exige, o desafio é tentar melhorar os resultados obtidos no capítulo anterior, quando se utilizou o procedimento de *Branch-and-Bound* do *IlogCplex 9.0* [13].

Na secção 4.2, depois de uma breve descrição generalista, desenvolve-se o uso da decomposição de Benders para tratamento do problema descrito no capítulo anterior com base na formulação $PLMC_4$ usando, em particular, um corte denominado de corte usual de Benders. Após um pequeno exemplo apresentam-se resultados obtidos considerando três variantes para a inicialização do método.

Na secção 4.3 e no sentido de tentar acelerar a convergência do método, será apresentada uma forma de obtenção de um corte potencialmente mais forte através da resolução de pequenos problemas de programação linear. Os resultados da experiência computacional serão apresentados considerando as três variantes de inicialização propostas na secção 4.2.

Na secção 4.4, propõe-se acelerar a convergência usando uma heurística como forma de obter um corte melhor que o corte usual. A ideia baseia-se no trabalho de Van Roy [49], que aplicou a técnica a um problema estático de localização com restrições de capacidade. No final da secção são apresentados os resultados obtidos a partir da experiência computacional.

Na secção 4.5, é aplicada a técnica de Benders com base na formulação $PLMC_3$. Propõe-se um reforço do corte usual, usando a heurística apresentada na secção 4.4. Duas estratégias são avaliadas tendo em conta uma pequena variante na heurística de reforço. Os resultados computacionais serão apresentados considerando as mesmas variantes para a inicialização do método já referidas atrás.

Por último, na secção 4.6, apresentam-se algumas conclusões e considerações acerca de todo o trabalho desenvolvido neste capítulo.

4.2 Aplicação da técnica de Benders

4.2.1 Breve introdução da técnica

A técnica da decomposição de Benders - também conhecida por decomposição primal (Magnanti e Wong [34], Van Roy [49]), por explorar a estrutura primal do modelo (embora fazendo uso de informação dual) - é usada na resolução de problemas que se complicam por um conjunto de variáveis inteiras, consideradas difíceis.

Benders [6] apresentou a técnica em 1962. Em 1974, Geoffrion and Graves [24], aplicam a técnica a um sistema de distribuição em larga escala. Em 1981, Magnanti e Wong [35] propõem uma forma para melhorar a *performance* da técnica. Também Van Roy [49] em 1986, contribui com um método para acelerar a convergência que passa por um fortalecimento dos cortes de Benders.

O uso da técnica difundiu-se então a problemas nas mais variadas áreas nomeadamente, na década de 90, na área das telecomunicações (Carey e Srinivasan [8], Luna *et al.* [43], Benchakroun *et al.* [5]), sistemas de produção e distribuição (Cordeau *et al.* [11]), sistemas de energia (Rafael *et al.* [42], Alguacil e Conejo [1], Hobbs *et al.* [31]), entre outros. Especificamente para problemas de localização, destacam-se os trabalhos de Van Roy [49] em 1986 que, como já foi referido, estuda um problema de localização estática com restrições de capacidade. Magnanti e Wong [34], analisam um problema de p-localização.

A pesquisa bibliográfica realizada no âmbito desta dissertação não permitiu encontrar qualquer aplicação desta técnica a problemas de localização multi-periódicos. Isso mesmo será feito neste capítulo.

Em termos gerais, a técnica consiste em dividir o conjunto de variáveis em dois subconjuntos: um com variáveis inteiras e outro com variáveis contínuas. Esta divisão produzirá uma divisão do problema original em dois subproblemas: um designado por problema mestre completo, PM, e outro designado por subproblema de Benders. Este último obtém-se quando são fixados os valores das variáveis inteiras e a sua resolução permite obter pontos extremos da sua região admissível, com os quais são construídas desigualdades lineares conhecidas como cortes de Benders ou cortes primais. De forma a evitar que se explicitem todos os pontos extremos, Benders propôs o uso de uma relaxação do problema mestre, PM_R, cuja resolução, em cada iteração do método, permite fixar as variáveis inteiras. Com esta informação, é então resolvido o subproblema dual que, por sua vez, propõe a PM_R uma correção dos valores das variáveis inteiras através do corte de Benders (Van Roy [49]). Esta restrição é adicionada a PM_R que, depois de resolvido, propõe uma nova configuração para o problema, mas refinada pela informação adicional fornecida pelo subproblema. Esta interactividade entre os problemas mantém-se até à convergência, obtendo-se no final, uma solução óptima para problema original. Benders demonstrou que este procedimento converge para a solução óptima do problema original.

Nesta secção detalha-se a aplicação da técnica da decomposição de Benders ao problema em estudo. A estrutura do problema sugere o uso desta técnica, uma vez que o conjunto de variáveis pode ser dividido em dois subconjuntos: o subconjunto das variáveis inteiras (localização) e o subconjunto das variáveis contínuas (afectação).

O problema resultante da fixação das variáveis z_{it} é um problema de programação linear, que pode ser decomposto em T problemas de transportes. De facto, em cada período do horizonte de planeamento, têm-se n origens, constituídas pelos serviços em funcionamento, e m destinos, representados pelos clientes. A disponibilidade em cada período é dada pela capacidade dos serviços em actividade, tendo que se satisfazer a procura total dos clientes nesse período.

4.2.2 Construção do corte de Benders e do problema mestre

Magnanti e Wong [34], salientam a importância da formulação que serve de base a um processo de decomposição. Duas formulações equivalentes de um mesmo problema, podem ter características computacionais muito diferentes, o que sugere algum cuidado na escolha do modelo a utilizar num procedimento de decomposição. Neste capítulo vai-se analisar a aplicação desta técnica tendo em conta as formulações $PLMC_3$ e $PLMC_4$. Nesta secção tratar-se-á a segunda e na secção 4.5 a primeira. Recorde-se que estas são as formulações que melhores resultados obtiveram no capítulo anterior e que resultaram da inclusão nos modelos $PLMC_1$ e $PLMC_2$ das desigualdades que garantem que a capacidade operacional em cada período é suficiente para satisfazer a procura. Aliás, este conjunto de restrições terá, como se verá mais adiante, um papel importante neste processo. Consideremos então o modelo $PLMC_4$, descrito na secção anterior:

$$\begin{aligned}
min \quad & \sum_{i\in I}\sum_{j\in J}\sum_{t\in TP} c_{ijt}x_{ijt} + \sum_{i\in I^c}\sum_{t=1}^{T-1} F_{it}z_{it} + \sum_{i\in I^c} F_{iT}\Big(1-\sum_{t=1}^{T-1} z_{it}\Big) + \sum_{i\in I^o}\sum_{t=2}^{T} F_{it}z_{it} && && (4.1)\\
s.a: \quad & \sum_{i\in I} x_{ijt} = 1 && j\in J,\ t\in TP && (4.2)\\
& x_{ijt} \le \Big(1-\sum_{\tau=1}^{t-1} z_{i\tau}\Big) && i\in I^c,\ j\in J,\ t\in TP && (4.3)\\
& x_{ijt} \le \sum_{\tau=2}^{t} z_{i\tau} && i\in I^o,\ j\in J,\ t\in TP && (4.4)\\
& \sum_{j\in J} d_{jt}x_{ijt} \le Q_i\Big(1-\sum_{\tau=1}^{t-1} z_{i\tau}\Big) && i\in I^c,\ t\in TP && (4.5)\\
& \sum_{j\in J} d_{jt}x_{ijt} \le Q_i\sum_{\tau=2}^{t} z_{i\tau} && i\in I^o,\ t\in TP && (4.6)\\
& \sum_{i\in I^c} Q_i\Big(1-\sum_{\tau=1}^{t-1} z_{i\tau}\Big) + \sum_{i\in I^o} Q_i\Big(\sum_{\tau=2}^{t} z_{i\tau}\Big) \ge \sum_{j\in J} d_{jt} && t\in TP && (4.7)\\
& \sum_{t\in TP} z_{it} \le 1 && i\in I && (4.8)\\
& x_{ijt} \ge 0 && i\in I,\ j\in J,\ t\in TP && (4.9)\\
& z_{it} \in \{0,1\} && i\in I,\ t\in TP && (4.10)
\end{aligned}$$

Os conjuntos (4.7), (4.8) e (4.10), só dependem das variáveis z_{it}, $i \in I$ e $t \in TP$. Vamos então definir o seguinte conjunto:

$$\mathbf{Z} = \{z_{it},\ i \in I, t \in TP : (4.7),\ (4.8)\ e\ (4.10)\ \text{são satisfeitas}\} \tag{4.11}$$

Repare-se que os conjuntos de restrições incluídos em $\mathbf{Z}$, são suficientes para produzirem uma configuração admissível para o problema original: por um lado, os conjuntos (4.8) e (4.10) garantem que o estado de cada um dos serviços não é alterado mais do que uma vez ao longo do horizonte de planeamento: por outro, o conjunto (4.7) assegura que, em cada um dos períodos, a capacidade total dos serviços em actividade é pelo menos o mínimo necessário para que a procura total seja satisfeita.

Ao fixarmos um vector $\mathbf{z} \in \mathbf{Z}$, o problema resultante é um problema que só depende das variáveis contínuas. Tal como já foi mencionado, este problema pode ser decomposto em T problemas de transportes. De facto, definida uma configuração, resta em cada período resolver o problema de afectação dos clientes aos serviços em funcionamento de forma a que a sua procura seja satisfeita na totalidade. Neste caso, como $\mathbf{z} \in \mathbf{Z}$, existe em cada período capacidade suficiente para satisfazer a procura total. Assim, cada um dos T problemas de transportes tem solução admissível.

O problema que resulta da fixação das variáveis inteiras é usualmente designado por subproblema primal e tem a seguinte forma:

$(SP_{\bar{z}})$

$$min \quad \sum_{i \in I} \sum_{j \in J} \sum_{t \in TP} c_{ijt} x_{ijt} + \sum_{i \in I^c} \sum_{t=1}^{T-1} F_{it} \bar{z}_{it} + \sum_{i \in I^c} F_{iT} \Big(1 - \sum_{t=1}^{T-1} \bar{z}_{it}\Big) + \sum_{i \in I^o} \sum_{t=2}^{T} F_{it} \bar{z}_{it} \tag{4.1'}$$

$s.a:$

$$\sum_{i \in I} x_{ijt} = 1 \qquad j \in J,\ t \in TP \tag{4.2}$$

$$x_{ijt} \le \Big(1 - \sum_{\tau=1}^{t-1} \bar{z}_{i\tau}\Big) \qquad i \in I^c,\ j \in J,\ t \in TP \tag{4.3'}$$

$$x_{ijt} \le \sum_{\tau=2}^{t} \bar{z}_{i\tau} \qquad i \in I^o,\ j \in J,\ t \in TP \tag{4.4'}$$

$$\sum_{j \in J} d_{jt} x_{ijt} \le Q_i \Big(1 - \sum_{\tau=1}^{T-1} \bar{z}_{i\tau}\Big) \qquad i \in I^c,\ t \in TP \tag{4.5'}$$

$$\sum_{j \in J} d_{jt} x_{ijt} \le Q_i \sum_{\tau=2}^{T} \bar{z}_{i\tau} \qquad i \in I^o,\ t \in TP \tag{4.6'}$$

$$x_{ijt} \ge 0 \qquad i \in I,\ j \in J,\ t \in TP \tag{4.9}$$

Note-se que se está a assumir que as variáveis z_{it}, $i \in I$ e $t \in TP$, estão fixadas.

Equivalentemente, tem-se:

$(SP_{\bar{z}})$

$$min \quad \sum_{i\in I}\sum_{j\in J}\sum_{t\in TP} c_{ijt}x_{ijt} + \sum_{i\in I^c}\sum_{t=1}^{T-1} F_{it}\bar{z}_{it} + \sum_{i\in I^c} F_{iT}\Big(1-\sum_{t=1}^{T-1}\bar{z}_{it}\Big) + \sum_{i\in I^o}\sum_{t=2}^{T} F_{it}\bar{z}_{it} \tag{4.1'}$$

$s.\,a:$

$$\sum_{i\in I} x_{ijt} = 1 \qquad j\in J,\ t\in TP \tag{4.2}$$

$$-\,x_{ijt} \geq -\Big(1-\sum_{\tau=1}^{t-1}\bar{z}_{i\tau}\Big) \qquad i\in I^c,\ j\in J,\ t\in TP \tag{4.3''}$$

$$-\,x_{ijt} \geq -\sum_{\tau=2}^{t}\bar{z}_{i\tau} \qquad i\in I^o,\ j\in J,\ t\in TP \tag{4.4''}$$

$$-\sum_{j\in J} d_{jt}x_{ijt} \geq -\,Q_i\,\Big(1-\sum_{\tau=1}^{t-1}\bar{z}_{i\tau}\Big) \qquad i\in I^c,\ t\in TP \tag{4.5''}$$

$$-\sum_{j\in J} d_{jt}x_{ijt} \geq -\,Q_i\,\sum_{\tau=2}^{t}\bar{z}_{i\tau} \qquad i\in I^o,\ t\in TP \tag{4.6''}$$

$$x_{ijt}\geq 0 \qquad i\in I,\ j\in J,\ t\in TP \tag{4.9}$$

Associando multiplicadores λ_{jt}, $j\in J$ e $t\in TP$, ao conjunto de restrições (4.2), π_{ijt}, $i\in I$, $j\in J$ e $t\in TP$, aos conjuntos (4.3") e (4.4") e v_{it}, $i\in I$ e $t\in TP$, aos conjuntos de restrições (4.5") e (4.6"), pode formular-se o dual linear de $SP_{\bar{z}}$ da seguinte forma:

$(DSP_{\bar{z}})$

$$\begin{aligned} max \quad & \sum_{i\in I^c}\sum_{t=1}^{T-1} F_{it}\bar{z}_{it} + \sum_{i\in I^c} F_{iT}\Big(1-\sum_{t=1}^{T-1}\bar{z}_{it}\Big) + \sum_{i\in I^o}\sum_{t=2}^{T} F_{it}\bar{z}_{it} + \sum_{j\in J}\sum_{t\in TP}\lambda_{jt} - \\ & \sum_{i\in I^c}\sum_{j\in J}\sum_{t\in TP}\pi_{ijt}\,\Big(1-\sum_{\tau=1}^{t-1}\bar{z}_{i\tau}\Big) - \sum_{i\in I^o}\sum_{j\in J}\sum_{t\in TP}\pi_{ijt}\,\sum_{\tau=2}^{t}\bar{z}_{i\tau} - \\ & \sum_{i\in I^c}\sum_{t\in TP} v_{it}\,Q_i\,\Big(1-\sum_{\tau=1}^{t-1}\bar{z}_{i\tau}\Big) - \sum_{i\in I^o}\sum_{t\in TP} v_{it}\,Q_i\,\sum_{\tau=2}^{t}\bar{z}_{i\tau} \end{aligned} \tag{4.12}$$

$s.\,a:$

$$\lambda_{jt} - \pi_{ijt} - d_{jt}\,v_{it} \leq c_{ijt} \qquad i\in I,\ j\in J,\ t\in TP \tag{4.13}$$

$$\pi_{ijt}\geq 0 \qquad i\in I,\ j\in J,\ t\in TP \tag{4.14}$$

$$v_{it}\geq 0 \qquad i\in I,\ t\in TP \tag{4.15}$$

$DSP_{\bar{z}}$ será designado por subproblema dual e V$(DSP_{\bar{z}})$ o seu valor óptimo, sendo o seu conjunto de soluções admissíveis representado por $\mathcal{F}(DSP_{\bar{z}})$.

$SP_{\bar{z}}$, como já foi mencionado, tem solução óptima finita, o que garante, para qualquer configuração admissível considerada, a obtenção de um ponto extremo de $\mathcal{F}(DSP_{\bar{z}})$. De facto, tratando-se de uma afectação parcial, cada cliente pode ser servido por mais do que um serviço em funcionamento. Como $\mathbf{z} \in \mathbf{Z}$, existe garantia de, em cada período, estar em actividade um número suficiente de serviços para satisfazer toda a procura (garantido pelo conjunto de restrições (4.7)). Então, existe $\bar{\boldsymbol{x}}$ tal que as restrições (4.2) são verificadas de tal forma que $\mathrm{V}(SP_{\bar{z}})$ é finito, pois $x_{ijt} \in [0,\ 1]$ e todos os custos considerados na função objectivo são finitos. Não há assim, nenhuma direcção ao longo da qual $\mathrm{V}(SP_{\bar{z}})$ possa tomar valores tão pequenos quanto queiramos.

Proposição 4.1 *$DSP_{\bar{z}}$ tem solução óptima finita de valor igual a $SP_{\bar{z}}$.* $\triangleleft$

Demonstração: Deriva do facto de $SP_{\bar{z}}$ ter solução óptima finita, verificando-se pelo, teorema da dualidade (forte), a igualdade $\mathrm{V}(SP_{\bar{z}}) = \mathrm{V}(DSP_{\bar{z}})$. ◀

Repare-se que $\mathcal{F}(DSP_{\bar{z}})$ não depende de $\mathbf{z}$ pelo que, para qualquer vector $\mathbf{z} \in \mathbf{Z}$ considerado, ao resolver o subproblema dual, obtém-se um ponto extremo da mesma região, $\mathcal{F}(DSP_{\bar{z}})$.

Podemos escrever $PLMC_4$ de forma compacta, nomeadamente:

$$\min_{\mathbf{z}\in\mathbf{Z}} \ \underbrace{\min_{\mathbf{x}\geq\mathbf{0}} \Big\{ (4.1) :\ (4.2),\ (4.3),\ (4.4),\ (4.5),\ (4.6)\ e\ (4.9) \text{ são satisfeitas } \Big\}}_{SP_z} \tag{4.16}$$

Dualizando o problema interno, SP_z, que sendo um problema de programação linear, pode ser substituído pela sua forma dual, transforma-se (4.16) num problema *minimax*, resultando na seguinte forma equivalente:

$$\min_{\mathbf{z}\in\mathbf{Z}} \ \Big\{ max\big\{ (4.12\,') : (\lambda,\ \pi,\ v) \in \Lambda\Pi V \big\} \Big\} \tag{4.17}$$

onde, $\Lambda\Pi V$ é o conjunto de todos os pontos extremos do conjunto das soluções admissíveis de DSP_z e $(4.12\,')$ designa a expressão (4.12) mas sem $\mathbf{z}$ fixo.

Linearizando (4.17) obtém-se:

$$\begin{aligned} &\min_{\mathbf{z}\in\mathbf{Z}} \ \rho \\ &\rho \geq (4.12') \qquad\qquad \forall\,(\lambda,\ \pi,\ v) \in \Lambda\Pi V \end{aligned} \tag{4.18}$$

A este problema dá-se o nome de Problema Mestre de Benders completo. A cada uma das desigualdades (4.18) dá-se o nome de corte de Benders, conhecido também por corte usual de Benders. Repare-se que este conjunto de restrições pode ser numeroso. No entanto, tendo em conta que nem todas as desigualdades contribuem para a definição da solução óptima podemos, na prática, prescindir de muitas delas. Benders propõe uma simplificação que passa por considerar uma relaxação do problema mestre, PM_R, em que se consideram apenas alguns cortes, em número suficiente para obter a convergência (McDaniel e Devine [36], Van Roy [49]).

Na notação original, o problema mestre relaxado, PM_R, toma a seguinte forma:

(PM_R)

$$
\begin{aligned}
& min \quad \rho \\
& s.a: \\
& \rho \geq \sum_{i \in I^c} \sum_{t=1}^{T-1} F_{it} z_{it} + \sum_{i \in I^c} F_{iT}\Big(1 - \sum_{t=1}^{T-1} z_{it}\Big) + \sum_{i \in I^o} \sum_{t=2}^{T} F_{it} z_{it} + \\
& \qquad \sum_{j \in J} \sum_{t \in TP} \lambda_{jt} - \sum_{i \in I^c} \sum_{t \in TP} \big(v_{it}\, Q_i + \sum_{j \in J} \pi_{ijt}\big) \Big(1 - \sum_{\tau=1}^{t-1} z_{i\tau}\Big) - \\
& \qquad \sum_{i \in I^o} \sum_{t \in TP} \big(v_{it}\, Q_i + \sum_{j \in J} \pi_{ijt}\big) \Big(\sum_{\tau=2}^{t} z_{i\tau}\Big) \qquad (\lambda,\, \pi,\, v) \in R \qquad (4.19) \\
& \sum_{i \in I^c} Q_i \Big(1 - \sum_{\tau=1}^{T-1} z_{i\tau}\Big) + \sum_{i \in I^o} Q_i \Big(\sum_{\tau=2}^{T} z_{i\tau}\Big) \geq \sum_{j \in J} d_{jt} \qquad t \in TP \qquad (4.7) \\
& \sum_{t \in TP} z_{it} \leq 1 \qquad i \in I \qquad (4.8) \\
& z_{it} \in \{0,1\} \qquad i \in I,\, t \in TP \qquad (4.10)
\end{aligned}
$$

em que $R \subseteq \Lambda\Pi V$. Equivalentemente,

(PM_R)

$$
\begin{aligned}
& \min_{\mathbf{z} \in \mathbf{Z}} \quad \sum_{i \in I^c} \sum_{t=1}^{T-1} F_{it} z_{it} + \sum_{i \in I^c} F_{iT}\Big(1 - \sum_{t=1}^{T-1} z_{it}\Big) + \sum_{i \in I^o} \sum_{t=2}^{T} F_{it} z_{it} + \rho \qquad (4.20) \\
& s.a: \\
& \rho \geq \sum_{j \in J} \sum_{t \in TP} \lambda_{jt} - \sum_{i \in I^c} \sum_{t \in TP} \big(v_{it}\, Q_i + \sum_{j \in J} \pi_{ijt}\big) \Big(1 - \sum_{\tau=1}^{t-1} z_{i\tau}\Big) - \\
& \qquad \sum_{i \in I^o} \sum_{t \in TP} \big(v_{it}\, Q_i + \sum_{j \in J} \pi_{ijt}\big) \Big(\sum_{\tau=2}^{t} z_{i\tau}\Big) \qquad (\lambda,\, \pi,\, v) \in R \qquad (4.21)
\end{aligned}
$$

Esta última formulação apresenta vantagens relativamente à anterior sobretudo a nível da linguagem de modelação utilizada (*IlogConcert 2.0* [12]) pois, em cada iteração, deixa de ser necessário construir a componente associada aos custos.

A inclusão dos conjuntos de restrições (4.7), (4.8) e (4.10) em PM_R, assegura, em cada iteração, a obtenção de uma configuração admissível.

Observação *Caso se considerasse* $\mathbf{z}$ *não admissível, o subproblema dual poderia ser ilimitado e nesse caso haveria a necessidade de adicionar outro tipo de cortes (cortes de admissibilidade), que envolvem as direcções extremas de* $\mathcal{F}(DSP_z)$*.* (Cordeau *et al.* [11])

$$\sum_{j\in J}\sum_{t\in TP}\lambda_{jt}-\sum_{i\in I^c}\sum_{t\in TP}v_{it}\,Q_i\left(1-\sum_{\tau=1}^{t-1}z_{i\tau}\right)-\sum_{i\in I^o}\sum_{t\in TP}v_{it}\,Q_i\sum_{\tau=2}^{t}z_{i\tau}-$$
$$\sum_{i\in I^c}\sum_{j\in J}\sum_{t\in TP}\pi_{ijt}\left(1-\sum_{\tau=1}^{t-1}z_{i\tau}\right)-\sum_{i\in I^o}\sum_{j\in J}\sum_{t\in TP}\pi_{ijt}\sum_{\tau=2}^{t}z_{i\tau}\leq 0,\qquad(\lambda,\,\pi,\,v)\in A\tag{4.22}$$

Em que A *é um subconjunto das direcções admissíveis caracterizadoras do conjunto de soluções admissíveis de* DSP_z*.* ♣

Em síntese, o algoritmo da decomposição de Benders começa com uma solução admissível para o problema. Em cada iteração o subproblema dual é resolvido fixando um vector $\mathbf{z}$. Com a solução óptima deste subproblema, é construído um corte de Benders que é adicionado a PM_R. Este é resolvido, gerando um novo vector $\mathbf{z}$ admissível. Prossegue-se assim por diante até se verificar o critério de paragem, isto é, quando o limite inferior e o melhor limite superior obtidos até ao momento estiverem suficientemente próximos.

Proposição 4.2 *Em cada iteração do algoritmo, o problema mestre relaxado produz um limite inferior e um limite superior para o valor óptimo do problema original.* $\triangleleft$

Demonstração: Ambos os problemas são de minimização e PM_R é uma relaxação de PM. Pelo que se verifica a desigualdade $\mathrm{V}(PM_R)\leq\mathrm{V}(PM)$. Como $\mathrm{V}(PM)=\mathrm{V}(PLMC_4)$, então

$$\mathrm{V}(PM_R)\leq\mathrm{V}(PLMC_4).$$

Por outro lado, os conjuntos de restrições (4.7), (4.8) e (4.10) presentes em PM_R são suficientes para que este produza uma configuração admissível para o problema original sendo o valor da solução admissível correspondente um limite superior para o valor óptimo do problema original. ◀

O facto de se ter disponível, em cada iteração, uma solução admissível, faz com que a decomposição de Benders possa ser usada como uma heurística (Van Roy [49]). Se se parar em qualquer iteração antes de ser atingida a solução óptima, tem-se disponível não só uma solução admissível para o problema como também um limite inferior que dá a possibilidade de medir a qualidade da melhor solução produzida durante o processo.

Proposição 4.3 *A sucessão de limites inferiores produzidos pelo problema mestre relaxado é uma sucessão não decrescente.* $\triangleleft$

Demonstração: O problema mestre é um problema de minimização. Sempre que é adicionado um corte de Benders, a nova região admissível está contida na região admissível anterior, pelo que o valor do problema é não inferior ao valor produzido no problema da iteração anterior. ◀

De forma a evitar que uma dada configuração seja considerada uma segunda vez, o que produziria um corte que não melhoraria o valor do problema mestre relaxado, foram introduzidos em cada iteração cortes de melhoramento, isto é, fazendo uso de informação do problema mestre anterior (Benoist *et al.* [7]). Assim, logo que disponível o primeiro limite inferior, em cada iteração, é adicionada ao problema mestre relaxado uma desigualdade do tipo

$$V(PM_R) > V(PM_R)_{anterior} \tag{4.23}$$

4.2.3 Formalização do algoritmo

Consideremos a seguinte notação adicional:

LI - limite inferior actual

LS - melhor limite superior obtido até ao momento

V(DSP_z) - valor do dual linear de SP_z

$nCortes$ - número de cortes de Benders considerados até ao momento

Observação *Embora os valores óptimos dos problemas mestre e original sejam iguais, a solução produzida pelo problema mestre é* $(\bar{z}, \rho)$ *e a solução do problema original é* $(\bar{z}, \bar{x})$*. Pelo que, o problema mestre fornece o valor óptimo e a configuração óptima (ou uma das óptimas, no caso de existirem soluções alternativas). Para determinar o valor das variáveis* x_{ijt}*, basta resolver o problema* SP_z*, fixando a configuração dada pelo problema mestre.* ♣

Estamos nas condições de apresentar formalmente o algoritmo da decomposição de Benders, quando aplicado ao problema de *phase-in/phase-out* multi-periódico para localização de serviços em estudo (algoritmo 4.1). Refira-se, desde já, que serão propostas três formas para inicialização do processo, ou seja, três alternativas para obtenção de uma solução admissível inicial.

Algorithm 4.1 Algoritmo da decomposição de Benders

Inicialização 1, 2 ou 3
Passo 1:
Resolver PM_R
$LI = V(PM_R)$
Actualizar o vector $\mathbf{z}$
Resolver DSP_z
$LS = min\{LS,\ V(DSP_z)\}$
Passo 2:
se $\frac{LS-LI}{LS} < \varepsilon$ ir para o **Passo 4**
Passo 3:
Construir o corte de Benders com a solução óptima de DSP_z, $(\lambda,\ \pi,\ v)$
Adicionar o corte a PM_R
$nCortes = nCortes + 1$
Voltar ao **Passo 1**
Passo 4:
Stop. A configuração associada ao valor de LS é a configuração óptima.

No algoritmo 4.1, ε representa uma quantidade arbitrariamente pequena.

Segundo Magnanti e Wong [34], vários estudos computacionais mostram que a selecção inicial dos cortes pode influenciar significativamente a *performance* do algoritmo. Aqueles autores sugerem que estes podem ser gerados com base em métodos heurísticos ou outros que produzam boas escolhas para as varáveis inteiras.

Tendo em conta que o vector $\mathbf{z}$ a fixar tem que ser admissível (de forma a garantir que DSP_z tem solução óptima finita) propõem-se, a seguir, três variantes para a inicialização do processo. A primeira é a que resulta de considerar a solução admissível obtida pelo procedimento heurístico apresentado na secção 3.3 (página 28).

Inicialização 1:

Obter uma solução admissível usando da heurística apresentada na secção 3.3. Seja V_H o seu valor.

$LI = -\infty$

$LS = V_H$

Resolver DSP_z usando o vector z que define a configuração admissível da heurística.

Construir o 1^o corte de Benders com a solução óptima de DSP_z e adicioná-lo a PM_R.

$nCortes = 1$

Naturalmente, que se o tempo de execução da heurística for elevado, esta inicialização pode não ser conveniente por consumir demasiado tempo de *CPU*. Como tal pode ser vantajoso começar com uma soluçao admissível trivial, fácil de obter e, portanto, com um esforço computacional baixo. Uma possibilidade é a solução que se obtém considerando todos os serviços a funcionar o máximo possível, ou seja, considerando a solução em que todos os serviços $i \in I^c$ estão em funcionamento ao longo de todo o horizonte de planeamento e todos os serviços $i \in I^o$ estão em funcionamento desde o período 2 até ao final do horizonte de planeamento (uma vez que só podem entrar em actividade a partir do período 2). Surge assim uma segunda inicialização para o algoritmo 4.1:

Inicialização 2:

Considerar a solução em que todos os serviços estão a funcionar o máximo possível. Seja V_S o seu valor:

$z_{it} = 0, \forall i \in I^c, t \in TP$

$z_{i2} = 1, \forall i \in I^o$

$z_{it} = 0, \forall i \in I^o, t = 3, ..., T$

$LI = -\infty$

Resolver DSP_z usando o vector z acima.

$LS = V(DSP_z)$

Construir o 1º corte de Benders com a solução óptima de DSP_z e adicioná-lo a PM_R.

$nCortes = 1$

A outra alternativa que aqui vai ser proposta para inicializar o algoritmo 4.1, resulta de começar por resolver o problema mestre sem qualquer corte. Recorde-se que o problema mestre tem restrições suficientes para garantir que a configuração obtida é admissível para o problema original. Como PM_R fornece um limite inferior para o problema, ter-se-á a seguinte inicialização:

Inicialização 3:

Resolver o PM_R sem qualquer corte.

$LI = V(PM_R)$

Resolver DSP_z sendo z o vector que representa a configuração obtida ao resolver PM_R.

$LS = V(DSP_z)$

Construir o 1º corte de Benders com a solução óptima de DSP_z e adicioná-lo a PM_R.

$nCortes = 1$

O processo iterativo definido no algoritmo 4.1 termina quando o problema mestre deixa de produzir soluções melhores que as conseguidas até ao momento. Tal pode acontecer, no pior caso, quando todas as configurações possíveis já foram testadas, mas na prática acontece muito antes. Como critério de paragem deste processo, considerou-se que é necessário gerar um número suficiente de cortes até se identificar uma solução com um desvio relativamente ao valor óptimo não superior a 0.1 %. O valor para a tolerância foi fixado com base no resultado de experiências computacionais preliminares que permitiram concluir que, só utilizando este valor ou inferior, se obtém uma configuração óptima para o problema original.

No entanto há quem defenda que, na prática, raramente se justifica a resolução até à optimalidade, uma vez que os dados (como os custos, procura e estimação das capacidades), usados no mundo real, contêm uma margem de erro superior a 1 % (Cordeau *et al.* [11]).

Vejamos uma aplicação do algoritmo da decomposição de Benders a uma pequena instância do problema. Saliente-se, desde já, que PM_R e os subproblemas, foram resolvidos utilizando o *general solver IlogCplex 9.0* [13] e a linguagem de modelação *IlogConcert 2.0* [12].

Exemplo 4.1 *Consideremos uma instância do problema de* phase-in/phase-out *multi-periódico que tem vindo a ser estudado, em que existem 5 possíveis localizações para os serviços. No inicío do horizonte de planeamento, composto por 4 períodos de tempo, já estão em funcionamento serviços nas localizações 1 e 2. Nas localizações 3, 4 e 5 podem ser instalados novos serviços. O número de clientes a servir em cada período é 10. Como condição de paragem considera-se* $\varepsilon = 0.001$.

Assuma-se uma capacidade associada a cada serviço dada por

i	1	2	3	4	5
Q_i	249	240	192	435	251

Como custos de operação, abertura e encerramento considerem-se os valores seguintes:

f_{it}	1	2	3	4
1	1719	1846	2022	5078
2	2821	2863	3052	4214
3	-	2016	2079	2096
4	-	1524	578	712
5	-	2350	2450	2639

h_{it}	1	2	3	4
3	-	3448	3595	682
4	-	3898	4198	84
5	-	4464	4818	206

g_{it}	1	2	3	4
1	1591	1703	620	-
2	2134	2322	277	-

Consideremos a procura dos clientes, em cada período, dada por:

d_{jt}	1	2	3	4
1	3	3	3	3
2	31	30	30	32
3	37	37	36	39
4	84	87	87	85
5	65	67	63	64
6	23	24	250	25
7	21	22	22	21
8	94	97	104	108
9	9	10	9	10
10	50	53	54	53

Cada uma das matrizes seguintes, representa o custo de satisfação da procura em cada um dos períodos.

c_{ij1}	1	2	3	4	5	6	7	8	9	10
1	4	108	265	287	776	116	113	1004	99	233
2	9	30	115	955	538	29	97	628	120	243

c_{ij2}	1	2	3	4	5	6	7	8	9	10
1	4	109	273	845	700	115	104	1035	102	225
2	10	27	108	982	515	29	88	607	125	246
3	29	228	156	1330	395	159	162	390	175	472
4	13	149	266	455	997	185	62	1102	62	71
5	33	272	210	1397	522	199	204	531	175	546

c_{ij3}	1	2	3	4	5	6	7	8	9	10
1	4	118	296	915	755	123	109	944	93	221
2	9	28	110	965	474	30	84	604	128	231
3	27	210	142	1332	434	158	167	353	160	516
4	12	144	284	472	902	186	66	1099	61	70
5	36	275	201	1401	493	214	202	568	159	560

c_{ij4}	1	2	3	4	5	6	7	8	9	10
1	4	116	311	999	757	112	109	925	87	217
2	9	26	113	926	492	32	81	594	141	208
3	26	111	141	224	215	143	175	172	169	158
4	12	33	11	25	56	69	64	72	58	75
5	38	293	218	1372	503	211	193	591	171	548

Inicialização 1:

A configuração admissível obtida quando se utiliza a heurística apresentada na secção 3.3 é a seguinte:

	1	2	3	4
1	■	□	□	□
2	■	■	■	■
3	□	□	□	□
4	□	■	■	■
5	□	□	□	□

custo = 30 080.66

$z_{11} = z_{42} = 1$

$e\ z_{it} = 0$ *para* $i \in I,\ t \in TP,\ (i,t) \neq (1,1)$ *e* $(i,t) \neq (4,2)$

LI = -∞
LS = 30 080.66

Para construir a função objectivo de PM_R *é conveniente construir-se a matriz* F_{it} *de acordo com (3.1), (3.2) e (3.3) (página 16),*

F_{it}	1	2	3	4
1	$g_{11} + f_{11}$	$g_{12} + f_{11} + f_{12}$	$g_{13} + f_{11} + f_{12} + f_{13}$	$f_{11} + f_{12} + f_{13} + f_{14}$
2	$g_{21} + f_{21}$	$g_{22} + f_{21} + f_{22}$	$g_{23} + f_{21} + f_{22} + f_{23}$	$f_{21} + f_{22} + f_{23} + f_{24}$
3	-	$h_{32} + f_{32} + f_{33} + f_{34}$	$h_{33} + f_{33} + f_{34}$	$h_{34} + f_{34}$
4	-	$h_{42} + f_{42} + f_{43} + f_{44}$	$h_{43} + f_{43} + f_{44}$	$h_{44} + f_{44}$
5	-	$h_{52} + f_{52} + f_{53} + f_{54}$	$h_{53} + f_{53} + f_{54}$	$h_{54} + f_{54}$

ou seja:

F_{it}	1	2	3	4
1	3 310	5 268	6 207	10 665
2	4 955	8 006	9 103	12 950
3	-	9 639	7 770	2 778
4	-	6 712	5 488	796
5	-	11 903	9 907	2 845

A função objectivo do problema mestre relaxado será então:

$$3\,310\, z_{11} + 5\,268\, z_{12} + 6\,207\, z_{13} + 10\,665\,(1 - z_{11} - z_{12} - z_{13}) + 4\,955\, z_{21} + 8\,006\, z_{22} + 9\,103\, z_{23} + 12\,950\,(1 - z_{21} - z_{22} - z_{23}) + 9\,639\, z_{32} + 7\,770\, z_{33} + 2\,778\, z_{34} + 6\,712\, z_{42} + 5\,488\, z_{43} + 796\, z_{44} + 11\,903\, z_{52} + 9\,907\, z_{53} + 2\,845\, z_{54} + \rho$$

Para além dos cortes a introduzir em cada iteração, as restrições de PM_R são:

$$249(1 - z_{11}) + 240(1 - z_{21}) + 192\, z_{32} + 435\, z_{42} + 251\, z_{52} \geq 430$$

$$249(1 - z_{11} - z_{12}) + 240(1 - z_{21} - z_{22}) + 192(z_{32} + z_{33}) + 435(z_{42} + z_{43}) + 251(z_{52} + z_{53}) \geq 658$$

$$249(1 - z_{11} - z_{12} - z_{13}) + 240(1 - z_{21} - z_{22} - z_{23}) + 192(z_{32} + z_{33} + z_{34}) + 435(z_{42} + z_{43} + z_{44}) + 251(z_{52} + z_{53} + z_{54}) \geq 440$$

$$z_{11} + z_{12} + z_{13} + z_{14} \leq 1$$

$$z_{21} + z_{22} + z_{23} + z_{24} \leq 1$$

$$z_{31} + z_{32} + z_{33} + z_{34} \leq 1$$

$$z_{41} + z_{42} + z_{43} + z_{44} \leq 1$$

$$z_{51} + z_{52} + z_{53} + z_{54} \leq 1$$

Resolvendo o subproblema dual com $z_{11} = z_{42} = 1$ e $z_{it} = 0$ para os restantes, obtém-se a seguinte solução:

λ_{jt}	1	2	3	4
1	4	13	10.872	9
2	108	149	46.72	26
3	208.096	266	132.464	11
4	787	455	472	25
5	701.54	997	513.312	56
6	86.87	185	186	32
7	113	62	66	64
8	864.516	1102	668.89	64
9	99	62	61	58
10	233	71	70	75

v_{it}	1	2	3	4
1	0	4.432	0.252	0
2	2.516	4.066	0.624	0
3	4.92	8.985	3.037	0
4	4.111	0	0	0
5	3.739	7.089	0.97	0

$\pi_{113} = 6.116$, $\pi_{114} = 5$, $\pi_{232} = 7.533$, $\pi_{252} = 209.533$, $\pi_{262} = 58.4$, $\pi_{282} = 100.533$, *com as restantes variáveis π_{ijt} iguais a 0.*

Com esta solução pode construir-se o primeiro corte a introduzir em PM_R:

$$\begin{aligned}
\rho \geq\ & 9214.296 - 0 \times 249 - (4.432 \times 249)(1 - z_{11}) - (0.252 \times 249 + 6.116)(1 - z_{11} - z_{12}) - \\
& (0 \times 249 + 5)(1 - z_{11} - z_{12} - z_{13}) - (2.516 \times 240) - (4.066 \times 240 + 375.999)(1 - z_{21}) - \\
& 0.624 \times 240(1 - z_{21} - z_{22}) - 0 \times 240(1 - z_{21} - z_{22} - z_{23}) - 8.985 \times 192\, z_{32} - \\
& 3.037 \times 192(z_{32} + z_{33}) - 0 \times 192(z_{32} + z_{33} z_{34}) - 0 \times 435\, z_{42} - 0 \times 435(z_{42} + z_{43}) - \\
& 0 \times 435(z_{42} + z_{43} + z_{44}) - 7.089 \times 251\, z_{52} - 0.97 \times 251(z_{52} + z_{53}) - 0 \times 251(z_{52} + z_{53} + z_{54})
\end{aligned}$$

$\Leftrightarrow$

$$\begin{aligned}
\rho \geq\ & 9214.296 - 1103.568(1 - z_{11}) - 68.864(1 - z_{11} - z_{12}) - 5(1 - z_{11} - z_{12} - z_{13}) - 603.84 - \\
& 1351.839(1 - z_{21}) - 149.76(1 - z_{21} - z_{22}) - 1725.12\, z_{32} - 583.1(z_{32} + z_{33}) - 1779.339\, z_{52} - \\
& 243.47(z_{52} + z_{53})
\end{aligned}$$

$\Leftrightarrow$

$$\begin{aligned}
\rho \geq\ & 5931.425 + 1177.432\, z_{11} + 73.864\, z_{12} + 5\, z_{13} + 1501.599\, z_{21} - \\
& 149.76\, z_{22} - 2308.22\, z_{32} - 583.1\, z_{33} - 2022.809\, z_{52} - 243.47\, z_{53}
\end{aligned}$$

nCortes = 1

Passo 1:

Resolvendo PM_R obtém-se a seguinte configuração:

	1	2	3	4	
1	■	■	■	□	$z_{13} = z_{21} = z_{34} = z_{42} = 1$
2	■	□	□	□	*e os restantes* $z_{it} = 0$
3	□	□	□	■	
4	□	■	■	■	*LI = V(PM_R) = 28 089.785*
5	□	□	□	□	

*Utilizando o vector **z** produzido pelo problema mestre, resolve-se DSP_z que dará não só o valor da solução mas também um novo vector (λ, π, v).*

V(DSP_z) = 29 551.81

LS = min{30 080.66, 29 551.81} = 29 551.81

λ_{jt}	1	2	3	4
1	4	13	4.756	15
2	108	149	125.56	65
3	208.096	266	284	50
4	787	455	472	110
5	701.54	997	770.876	120
6	86.87	185	186	94
7	113	62	66	85
8	864.516	1102	970.2	172
9	99	62	61	68
10	233	71	70	128

v_{it}	1	2	3	4
1	0	0	0.252	0
2	2.516	7.194	4.833	2.48
3	4.92	8.985	5.934	0
4	4.111	0	0	1
5	3.739	7.089	4.41	0

$\sum_{j\in J}\sum_{t\in TP}\lambda_{jt}$=*10 484.432*

$\pi_{112} = 9$, $\pi_{114} = 11$, $\pi_{122} = 40$, $\pi_{152} = 297$, $\pi_{162} = 70$, $\pi_{182} = 67$, *sendo as restantes variáveis* π_{ijt} *iguais a 0.*

Passo 2:

Critério de paragem: $\frac{LS-LI}{LS} = \frac{29551.81-28089.785}{29551.81} = 0.04947 > 0.001$

Passo 3:

Com a solução de DSP_z *podemos construir o segundo corte:*

$$\rho \geq 10484.432 - 483(1-z_{11}) - 62.748(1-z_{11}-z_{12}) - 11(1-z_{11}-z_{12}-z_{13}) - 603.84 - 1726.56(1-z_{21}) - 1159(1-z_{21}-z_{22}) - 595.2(1-z_{21}-z_{22}-z_{23}) - 1725.12\, z_{32} - 1139.328(z_{32}+z_{33}) - 435(z_{42}+z_{43}+z_{44}) - 1779.339\, z_{52} - 1106.91(z_{52}+z_{53})$$

$\Leftrightarrow$

$$\rho \geq 5843.084 + 556.748\, z_{11} + 74.748\, z_{12} + 11\, z_{13} + 3480.76\, z_{21} + 1754.2\, z_{22} + 595.2 z_{23} - 2864.448\, z_{32} - 1139.328\, z_{33} - 435\, z_{42} - 435\, z_{43} - 435\, z_{44} - 2886.249\, z_{52} - 1106.91\, z_{53}$$

nCortes = 2

Passo 1:

Introduz-se em PM_R *o corte obtido no **Passo 3** e resolve-se o novo problema mestre relaxado. A configuração obtida é a seguinte:*

	1	2	3	4
1	■	■	□	□
2	■	■	■	□
3	□	□	□	■
4	□	□	■	■
5	□	□	□	□

$z_{12} = z_{23} = z_{34} = z_{43} = 1$
e para os restantes $z_{it} = 0$

$LI = V(PM_R) = 28\,622.994$

Resolvendo DSP_z *obtém-se o seu valor que traduz o valor da solução associada à configuração fornecida por* PM_R*.*

V(DSP_z) = *30 304.665*

LS = *min*{*29 551.81, 30 304.665*} = *29 551.81*

λ_{jt}	1	2	3	4
1	4	4	10.872	15
2	108	109	46.72	65
3	208.096	209.133	132.464	50
4	787	845	472	110
5	701.54	698.133	513.312	120
6	86.87	94.6	186	94
7	113	104	66	85
8	864.516	872.133	668.896	172
9	99	102	61	68
10	233	225	70	128

v_{it}	1	2	3	4
1	0	0	0.252	0
2	2.516	2.733	0.624	2.48
3	4.92	4.97	3.037	0
4	4.111	4.482	0	1
5	3.739	3.516	0.97	0

$$\sum_{j\in J}\sum_{t\in TP}\lambda_{jt}{=}9\ 602.296$$

$\pi_{113} = 6.116$, $\pi_{114} = 11$, *sendo os restantes* π_{ijt} *iguais a 0.*

Passo 2:

Critério de paragem: $\frac{LS-LI}{LS} = \frac{29551.81-28622.994}{29551.81} = 0.03143 > 0.001$

Passo 3:

Constrói-se o corte associado a esta solução, resultando em:

$$\rho \geq 9602.296 - 68.864(1 - z_{11} - z_{12}) - 11(1 - z_{11} - z_{12} - z_{13}) -$$
$$655.92(1 - z_{21}) - 149.76(1 - z_{21} - z_{22}) - 595.2(1 - z_{21} - z_{22} - z_{23}) - 954.24\, z_{32} -$$
$$583.104(z_{32} + z_{33}) - 1949.67\, z_{42} - 435(z_{42} + z_{43} + z_{44}) - 882\, z_{52} - 243.47(z_{52} + z_{53})$$

$\Leftrightarrow$

$$\rho \geq 8711.552 + 79.864\, z_{11} + 79.864\, z_{12} + 11\, z_{13} + 1400.88\, z_{21} + 744.96\, z_{22} + 595.2\, z_{23} -$$
$$1537.344\, z_{32} - 583.104\, z_{33} - 2384.67\, z_{42} - 435\, z_{43} - 435\, z_{44} - 1125.47\, z_{52} - 243.47\, z_{53}$$

nCortes = *3*

Passo 1:

Depois de introduzido em PM_R *o corte acima, resolve-se este último problema sendo a configuração obtida a seguinte:*

	1	2	3	4
1	■	□	□	□
2	■	■	■	□
3	□	□	□	■
4	□	■	■	■
5	□	□	□	□

$z_{11} = z_{23} = z_{34} = z_{42} = 1$
e para os restantes $z_{it} = 0$

$LI = V(PM_R) = 28\ 921.665$

Resolve-se DSP_z *obtendo-se:*

$V(DSP_z) = 28\ 973.665$

$LS = min\{29\ 551.81,\ 28\ 973.665\} = 28\ 973.665$

λ_{jt}	1	2	3	4
1	4	13	10.872	15
2	108	149	46.72	65
3	208.096	266	132.464	50
4	787	455	472	110
5	701.54	997	513.312	120
6	86.87	185	186	94
7	113	62	66	85
8	864.516	1102	668.896	172
9	99	62	61	68
10	233	71	70	128

v_{it}	1	2	3	4
1	0	4.432	0.252	0
2	2.516	4.066	0.624	2.48
3	4.92	8.985	3.037	0
4	4.111	0	0	1
5	3.739	7.089	0.97	0

$$\sum_{j\in J}\sum_{t\in TP}\lambda_{jt}=9\ 701.296$$

$\pi_{113} = 6.116,\ \pi_{114} = 11,\ \pi_{232} = 7.533,\ \pi_{252} = 209.533,\ \pi_{262} = 58.4,\ \pi_{282} = 100.533$ *sendo os restantes* π_{ijt} *iguais a 0.*

Passo 2:

Critério de paragem: $\frac{LS-LI}{LS} = \frac{28973.665-28921.665}{28973.665} = 0.001794 > 0.001$

Passo 3:

O corte gerado pela solução de DSP_z *solução é o seguinte:*

$$\rho \geq 9701.296 - 1103.568(1-z_{11}) - 68.864(1-z_{11}-z_{12}) - 11(1-z_{11}-z_{12}-z_{13}) - 603.84 - 1351.839(1-z_{21}) - 149.76(1-z_{21}-z_{22}) - 595.2(1-z_{21}-z_{22}-z_{23}) - 1725.12\, z_{32} - 583.104(z_{32}+z_{33}) - 435(z_{42}+z_{43}+z_{44}) - 1779.339\, z_{52} - 243.47(z_{52}+z_{53})$$

$\Leftrightarrow$

$$\rho \geq 5817.225 + 1183.432\, z_{11} + 79.864\, z_{12} + 11\, z_{13} + 2096.799\, z_{21} + 744.96\, z_{22} + 595.2\, z_{23} - 2308.224\, z_{32} - 583.104\, z_{33} - 435\, z_{42} - 435\, z_{43} - 435\, z_{44} - 2022.809\, z_{52} - 243.47\, z_{53}$$

$nCortes = 4$

Passo 1:

O corte é introduzido em PM_R que depois de resolvido conduz à seguinte configuração:

	1	*2*	*3*	*4*
1	■	□	□	□
2	■	■	■	□
3	□	□	□	■
4	□	■	■	■
5	□	□	□	□

$z_{11} = z_{23} = z_{34} = z_{42} = 1$
e para os restantes $z_{it} = 0$

$LI = V(PM_R) = 28\ 973.665$

Resolve-se DSP_z obtendo-se o valor da solução associada à configuração dada por PM_R.

$V(DSP_z) = 28\ 973.665$

$LS = min\{28\ 973.665,\ 28\ 973.665\} = 28\ 973.665$

Passo 2:

Critério de paragem: $\frac{LS-LI}{LS} = \frac{28973.665-28973.665}{28973.665} = 0 < 0.001 \Rightarrow$ ***Passo 4***

Passo 4

*Stop. A configuração dada por PM_R é óptima para o problema original. Para obter o valor das variáveias x_{ijt} basta resolver o subproblema primal com o vector **z** fixo.* $\diamondsuit$

4.2.4 Experiência computacional

Apresentam-se nesta subsecção, os resultados obtidos através da aplicação da técnica da decomposição de Benders, e tendo em conta as três variantes de inicialização: com a solução da heurística apresentada na secção 3.3; considerando todos os serviços em funcionamento; resolvendo inicialmente o problema mestre relaxado sem qualquer corte.

Para cada variante, apresenta-se o tempo de *CPU* (em segundos) e o número de cortes gerados até à convergência. Os testes foram efectuados sobre o mesmo conjunto de instâncias usadas nos testes computacionais do capítulo anterior. Como já foi referido anteriormente, na resolução de PM_R e DSP_z foi sempre utilizado o *general solver IlogCplex 9.0* [13] e usando a linguagem de modelação *IlogConcert 2.0* [12].

Para as instâncias em que se consideram 10 possíveis localizações, os resultados constam na tabela 4.1. Nas tabelas 4.2 e 4.3, apresentam-se os resultados obtidos para as instâncias com 20 e 50 localizações, respectivamente.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		3	0.13	1	0.06	1	0.08	1
5		5	0.39	5	0.36	5	0.34	5
		8	0.27	3	0.20	3	0.23	3
		3	0.89	5	0.75	5	0.92	5
10	20	5	1.45	5	1.27	5	1.11	5
		8	1.44	4	1.20	4	1.31	4
		3	186.82	13	213.19	13	349.69	14
15		5	12588.41	30	13649.2	27	15829.78	28
		8	18.97	9	15.77	9	21.24	12
		3	1.05	2	0.22	2	0.22	2
5		5	1.36	3	0.34	3	0.36	3
		8	2.08	7	0.76	7	0.81	7
		3	7.84	5	4.07	5	4.45	4
10	50	5	4.60	4	0.95	4	0.98	4
		8	14.16	14	20.16	15	18.47	16
		3	351.52	20	298.78	18	163.25	14
15		5	248.57	14	237.12	14	187.08	13
		8	1241.21	29	871.80	26	1271.56	29
		3	3.09	4	0.61	4	0.66	4
5		5	3.11	1	0.25	3	0.44	2
		8	3.81	5	0.91	5	0.95	5
		3	14.84	9	5.08	9	5.14	9
10	100	5	18.19	10	10.27	10	10.24	10
		8	12.84	7	4.02	7	1.02	7
		3	129.59	9	104.34	9	106.87	9
15		5	2823.58	24	2802.11	24	2028.27	27
		8	95.50	14	60.12	14	67.89	14

Tabela 4.1: Tempo de *CPU* (seg) e *#cortes* - 10 localizações.

T	#J	$\#I^c$	c/ Sol. Heur *tempo*	*#cortes*	c/ Serv. abertos *tempo*	*#cortes*	s/ cortes inic. *tempo*	*#cortes*
		8	9.89	2	0.42	2	0.41	2
5		10	7.59	2	0.41	2	0.42	2
		16	16.80	28	5.55	28	5.91	26
		8	487.89	18	491.48	20	557.78	18
10	50	10	1084.92	25	1303.67	33	1680.36	31
		16	217.47	18	277.44	15	226.94	19
		8	>12 h	-	>12 h	-	>12 h	-
15		10	18515.43	63	27807.05	54	17790.27	53
		16	5432.15	25	3443.75	26	5895.63	28
		8	38.69	5	1.82	5	2.36	6
5		10	27.92	3	1.38	3	1.05	2
		16	38.47	30	17.40	32	15.15	27
		8	456.63	26	209.90	24	189.01	26
10	100	10	215.30	23	73.13	23	67.28	21
		16	470.30	37	289.45	38	361.29	43
		8	6771.15	30	4648.83	34	6752.56	32
15		10	> 12h	-	>12h	-	>12h	-
		16	3832.11	23	3606.08	23	4274.82	21

Tabela 4.2: Tempo de *CPU* (seg) e *#cortes* - 20 localizações.

T	#J	$\#I^c$	c/ Sol. Heur *tempo*	*#cortes*	c/ Serv. abertos *tempo*	*#cortes*	s/ cortes inic. *tempo*	*#cortes*
		20	90.91	4	6.47	4	6.87	4
5		25	101.41	5	8.52	5	9.95	6
		45	161.88	26	66.15	30	80.98	37
		20	5690.91	37	11209.83	25	2709.79	32
10	100	25	8734.11	27	14849.45	34	11273.67	27
		45	819.36	22	296.22	22	208.42	28
		20	843.49	4	55.83	3	216.06	3
15		25	*out mem.*	-	7.52	1	*out mem.*	-
		45	1738.81	16	2600.04	16	1345.30	18

Tabela 4.3: Tempo de *CPU* (seg) e *#cortes* - 50 localizações.

⌜ *Para a instância assinalada com 'out mem.' na tabela 4.3, o solver apresentou um erro de memória aquando da resolução de* PM_R. ⌟

A tabela 4.4 apresenta os valores médios associados para as instâncias com 10 localizações, usando cada uma das 3 variantes de inicialização.

variante	tempo	cortes
Inicialização 1	658.36	9.5
Inicialização 2	677.92	9.3
Inicialização 3	743.46	9.5

Tabela 4.4: Valores médios - 10 localizações.

Para este grupo de instâncias e de acordo com o resumo da tabela, verifica-se que usando a segunda variante de inicialização, foram gerados, em média, menos cortes do que nas outras duas variantes. Contudo, as diferenças não são significativas. O tempo médio de *CPU* consumido quando usada a segunda inicialização é superior ao tempo médio quando usada a primeira variante de inicialização, devendo-se esta diferença à instância que considera 15 períodos de tempo, 20 clientes e $\#I^c = 5$. Observando a tabela 4.1, verifica-se que em 24 das 27 instâncias, o tempo gasto pelo algoritmo quando inicializado com a segunda variante, é inferior ao tempo gasto quando usada a variante 1 e em 18 delas é inferior à variante 3. Ainda de acordo com a tabela 4.1, a incialização 2 é a que apresenta melhor *performance* para este grupo de instâncias que, mesmo tratando-se de uma solução trivial, gera, na grande maioria das instâncias, igual ou menor número de cortes na resolução de cada uma delas. Verifica-se ainda que o aumento do número de períodos do horizonte de planeamento é mais crítico que o aumento, mesmo que substancial, do número de clientes. Pode ainda constatar-se que considerar um número de serviços existentes no início do horizonte de planeamento igual ou superior a $\frac{\#I}{2}$ aumenta, também, a dificuldade de resolução, nomeadamente quando T=15.

A tabela que se segue apresenta informação semelhante à tabela 4.4 considerando, agora, as instâncias com 20 localizações. No entanto, devido ao facto de existirem duas instâncias para as quais o algoritmo da decomposição não obteve a solução óptima em tempo inferior a 12 horas (43 200 segundos) usando qualquer das variantes de inicialização, as mesmas não serão consideradas na tabela dos valores médios. Deve referir-se que o tempo de resolução de cada problema mestre relaxado foi, para estas instâncias, especialmente elevado.

variante	tempo	cortes
Inicialização 1	2351.42	22.4
Inicialização 2	2636.11	22.6
Inicialização 3	2363.83	22.3

Tabela 4.5: Valores médios - 20 localizações.

Tendo em conta o conjunto de instâncias com 20 localizações e considerando apenas as 16 instâncias para as quais se conseguiu uma solução em tempo inferior a 12 horas e com base na tabela 4.5, podemos verificar que a variante de inicialização 1 conduz a um tempo médio de *CPU* inferior a qualquer das restantes variantes, sendo a diferença muito ligeira em relação à variante 3. O número médio de cortes gerados, quando usada cada uma das variantes de inicialização, é muito equilibrado. No entanto, observando a tabela 4.2, verifica-se que a primeira variante de inicialização, embora gerando por vezes menor número de cortes (mas não substancialmente menor), acaba por consumir mais tempo. Tal permite concluir que o tempo gasto na obtenção de uma solução admissível não é compensado pelo facto de se ter disponível uma boa escolha para o vector z.

Em mais de metade das instâncias deste grupo, a segunda variante gerou igual ou menor número de cortes que a primeira variante. O tempo de resolução de cada uma das instâncias é, na maioria dos casos, significativamente inferior. De realçar ainda o facto de neste grupo serem as intâncias que consideram um número de serviços existentes igual ou inferior a $\frac{\#I}{2}$ as que apresentaram maior dificuldade na resolução, especialmente quando T=15.

A tabela 4.6 apresenta os valores médios correspondentes a 8 das 9 instâncias com 50 localizações, uma vez que se excluiu do cálculo de médias a instância para a qual o *general solver* deu um erro de memória.

variante	tempo	cortes
Inicialização 1	2272.6	17.6
Inicialização 2	3636.56	17.4
Inicialização 3	1981.38	19.4

Tabela 4.6: Valores médios - 50 localizações.

Este é o grupo de instâncias com maior dimensão. Isto traduz-se num elevado número de cortes necessário para obter a solução óptima e, consequentemente, num elevado número de problemas mestre relaxados a resolver considerando, em cada um, um elevado número de variáveis inteiras. A variante de inicialização 3 foi a que conduziu a menor tempo médio de CPU mesmo tendo sido a que gerou, em média, maior número de cortes. A variante de inicialização 1 teve, neste grupo de instâncias, um melhor desempenho do que a variante 2. De acordo com a tabela 4.3, para as instâncias com 10 períodos de tempo, especialmente as que consideram um número de serviços existentes no início do horizonte de planeamento não superior a $\frac{\#I}{2}$ e para qualquer das variantes, foram produzidos tempo e número de cortes muito elevados.

Na secção seguinte tentar-se-á ultrapassar algumas dificuldades introduzindo, em cada iteração, um corte potencialmente mais forte obtido a partir do corte usual. Com a utilização de um corte que à partida poderá ser mais forte, poder-se-á reduzir o número de cortes necessário para alcançar uma solução óptima. Assim, o número de problemas mestre relaxados a resolver será menor e, consequentemente, o tempo de CPU necessário para obter a solução óptima poderá diminuir.

4.3 Reforço do corte usual

Num processo de decomposição está naturalmente implícita a redução da dimensão dos problemas a resolver. Contudo, a decomposição de Benders é conhecida por ser um procedimento que, em geral, requer um elevado esforço computacional. O número de cortes necessário para obter a convergência pode ser muito elevado, além de que o facto de, em cada iteração, ser necessário resolver um problema de programação linear inteira, dificulta ainda mais a tarefa. Estes aspectos ficaram de alguma forma evidenciados nos resultados obtidos na secção anterior.

Uma das formas de atenuar este problema consiste em aumentar a qualidade do corte a introduzir em cada iteração, o que pode influenciar significativamente a *performance* do algoritmo (Magnanti e Wong [34], Van Roy [49]). O objectivo nesta secção é tentar melhorar a qualidade dos cortes, de forma a reduzir os tempos produzidos pelo algoritmo da decomposição de Benders quando usado o corte usual.

4.3.1 Metodologia

Atendendo a que o subproblema primal, SP_z, é um problema que pode ser de decomposto em T problemas de transportes, é natural a multiplicidade de soluções óptimas do seu dual, DSP_z. Desta forma, poderá ser possível obter mais do que um corte de Benders, podendo, uma determinada solução dual óptima originar um corte mais forte que outra. Na prática, para identificar um corte mais forte, pode verificar-se se nas restrições de corte, existe uma em que o valor do membro direito da restrição é maior para qualquer valor de z_{it} (Magnanti e Wong [34]). Assim, a partir do corte usual, podemos tentar obter um corte mais forte aumentando o seu membro direito.

O corte usual, apresentado na secção anterior, tem a forma:

$$\rho \geq \sum_{j\in J}\sum_{t\in TP}\lambda_{jt} - \sum_{i\in I^c}\sum_{t\in TP}\big(v_{it}\,Q_i + \sum_{j\in J}\pi_{ijt}\big)\Big(1-\sum_{\tau=1}^{t-1} z_{i\tau}\Big) - \sum_{i\in I^o}\sum_{t\in TP}\big(v_{it}\,Q_i + \sum_{j\in J}\pi_{ijt}\big)\Big(\sum_{\tau=2}^{t} z_{i\tau}\Big) \tag{4.21'}$$

Uma forma de aumentar o lado direito da desigualdade (4.21'), consiste em tentar diminuir os coeficientes $\big(v_{it}\,Q_i + \sum_{j\in J}\pi_{ijt}\big)$, $i\in I$ e $t\in TP$.

Fixemo-nos numa configuração e consideremos o corte usual por ela induzido. Para $i\in I^c$, $t\in TP$, se $\Big(1-\sum_{\tau=1}^{t-1} z_{i\tau}\Big) = 0$ (isto é, se $\sum_{\tau=1}^{t-1} z_{i\tau} = 1$), o coeficiente associado (ou seja $v_{it}\,Q_i + \sum_{j\in J}\pi_{ijt}$) pode ser modificado sem que a função objectivo de PM_R sofra alteração, desde que se mantenha a admissiblidade do dual. Tal consegue-se resolvendo o seguinte problema de programação linear:

(PR_{it})

$$\min_{v_{it},\pi_{ijt}\geq 0} \quad v_{it}\, Q_i \; + \sum_{j\in J} \pi_{ijt} \tag{4.24}$$

$$s.\,a:$$

$$\lambda_{jt} - \pi_{ijt} - d_{jt}\, v_{it} \leq c_{ijt} \qquad\qquad j \in J \tag{4.25}$$

Uma análise mais aprofundada ainda permite concluir que se para um dado par $(i,\ t)$, $\Big(1 - \sum_{\tau=1}^{t-1} z_{i\tau}\Big) = 0$, então existe a possibilidade de alterar outras variáveis para além de v_{it} e π_{ijt}, $j \in J$. Concretamente, será possível alterar as variáveis $v_{i,t+1}, \cdots, v_{iT}$ e $\pi_{ij\tau}$, $j \in J$, $\tau = t{+}1, \cdots, T$. Vejamos que assim é. Seja $i \in I^c$. Tem-se

$$\begin{aligned}
&\sum_{t\in TP} \big(v_{it}\, Q_i + \sum_{j\in J} \pi_{ijt}\big)\Big(1 - \sum_{\tau=1}^{t-1} z_{i\tau}\Big) = \\
&\big(v_{i1}\, Q_i + \sum_{j\in J} \pi_{ij1}\big) + \\
&\big(v_{i2}\, Q_i + \sum_{j\in J} \pi_{ij2}\big)\big(1 - z_{i1}\big) + \\
&\big(v_{i3}\, Q_i + \sum_{j\in J} \pi_{ij3}\big)\big(1 - z_{i1} - z_{i2}\big) + \\
&\big(v_{i4}\, Q_i + \sum_{j\in J} \pi_{ij4}\big)\big(1 - z_{i1} - z_{i2} - z_{i3}\big) + \\
&\qquad\qquad \vdots \\
&\big(v_{it}\, Q_i + \sum_{j\in J} \pi_{ijt}\big)\big(1 - z_{i1} - z_{i2} - \ldots - z_{i,t-1}\big) + \\
&\qquad\qquad \vdots \\
&\big(v_{iT}\, Q_i + \sum_{j\in J} \pi_{ijT}\big)\big(1 - z_{i1} - z_{i2} - \ldots - z_{i,t} - \ldots - z_{i,T-1}\big)
\end{aligned}$$

Se $z_{i1} = 1$, podem alterar-se os valores das variáveis v_{i2}, v_{i3}, $\ldots v_{iT}$ e π_{ijt}, $j \in J$, $t = 2, \cdots, T$.

Se $z_{i2} = 1$, podem alterar-se os valores das variáveis v_{i3}, v_{i4}, $\ldots v_{iT}$ e π_{ijt}, $j \in J$, $t = 3, \cdots, T$.

Em geral,

Se $z_{i,t} = 1$, podem ser alteradas as variáveis $v_{i,t+1}$, $\ldots v_{iT}$ e $\pi_{ij\tau}$, $j \in J$, $\tau = t + 1, \cdots, T$.

O mesmo tipo de análise pode ser feita para os coeficientes associados a $\big(\sum_{\tau=2}^{t} z_{i\tau}\big)$, $i \in I^o$ e $t \in TP$.

Se $\big(\sum_{\tau=2}^{t} z_{i\tau}\big) = 0$, pode diminuir-se o respectivo coeficiente resolvendo o problema linear PR_{it} correspondente.

Vejamos quais as variáveis v_{it} e π_{ijt} correspondentes que poderão ser alteradas. Seja $i \in I^o$.

$$
\begin{aligned}
&\sum_{t \in TP} \big(v_{it}\, Q_i + \sum_{j \in J} \pi ijt\big) \Big(\sum_{\tau=2}^{t} z_{i\tau}\Big) = \\
&\big(v_{i2}\, Q_i + \sum_{j \in J} \pi_{ij2}\big)\, z_{i2} + \\
&\big(v_{i3}\, Q_i + \sum_{j \in J} \pi_{ij3}\big)\big(z_{i2} + z_{i3}\big) + \\
&\big(v_{i4}\, Q_i + \sum_{j \in J} \pi_{ij4}\big)\big(z_{i2} + z_{i3} + z_{i4}\big) + \\
&\big(v_{i5}\, Q_i + \sum_{j \in J} \pi_{ij5}\big)\big(z_{i2} + z_{i3} + z_{i4} + z_{i5}\big) + \\
&\qquad\qquad\qquad\vdots \\
&\big(v_{it}\, Q_i + \sum_{j \in J} \pi_{ijt}\big)\big(z_{i2} + z_{i3} + \ldots + z_{i,t}\big) + \\
&\qquad\qquad\qquad\vdots \\
&\big(v_{iT}\, Q_i + \sum_{j \in J} \pi_{ijT}\big)\big(z_{i2} + z_{i3} + \ldots + z_{i,t} + \ldots + z_{iT}\big)
\end{aligned}
$$

Por um lado, se $z_{i2} + z_{i3} + \ldots + z_{iT} = 0$, então podem ser alteradas as variáveis v_{i2}, v_{i3}, $\ldots$, v_{iT} e π_{ijt}, $j \in J$, $t = 2, \cdots, T$.

Por outro lado, se $z_{it} = 1$ para algum par $(i,\, t)$ então $z_{i2} + z_{i3} + \ldots + z_{i,t-1} = 0$, o que significa que podem ser alteradas as variáveis v_{i2}, v_{i3}, $\ldots$, $v_{i,t-1}$ e π_{ijt}, $j \in J$, $t = 2, \cdots, t-1$.

O algoritmo 4.2 sistematiza as considerações feitas atrás. São identificados os pares (i, t) associados a possíveis alterações de forma a que o corte anterior possa ser fortalecido.

Algorithm 4.2 Procedimento de fortalecimento do corte

for $i \in I^c$ **do**
  Se existir τ tal que $z_{i\tau} = 1$
  for t $= \tau + 1, \ldots, T$ **do**
    Resolver o problema PR_{it} para o par (i, t)
for $i \in I^o$ **do**
  if $\sum_{\tau=2}^{T} z_{i\tau} = 0$ **then**
    for $t = 2, \ldots, T$ **do**
      Resolver o problema PR_{it} para o par (i, t)
  else
    Determinar τ tal que $z_{i\tau} = 1$
    for $t = 2, \ldots, \tau - 1$ **do**
      Resolver o problema PR_{it} para o par (i, t)

O algoritmo da decomposição de Benders com corte fortalecido toma a seguinte forma:

Algorithm 4.3 Algoritmo da decomposição de Benders com fortalecimento do corte

Inicialização 1, 2 ou 3
Passo 1:
Resolver PM_R
$LI = V(PM_R)$
Actualizar o vector **z**
Resolver DSP_z
$LS = \min\{LS, V(DSP_z)\}$
Passo 2:
se $\frac{LS-LI}{LS} < \varepsilon$ ir para o **Passo 4**
Passo 3:
Executar o algoritmo (4.2)
Construir o corte de Benders
Adicionar o corte a PM_R
$nCortes = nCortes + 1$
Voltar ao **Passo 1**
Passo 4:
Stop. A configuração associada ao valor de LS é a configuração óptima.

Novamente, ε representa uma quantidade arbitrariamente pequena.

Com esta forma de construir o corte, é necessária uma ligeira alteração em cada uma das variantes de inicialização, bastando que a seguir à resolução de DSP_z se aplique o algoritmo (4.2):

Inicialização 1:

Obter uma solução admissível usando da heurística apresentada na secção 3.3 e seja V_H o seu valor.

$LI = -\infty$

$LS = V_H$

Resolver DSP_z

Executar o algoritmo (4.2)

Construir o corte de Benders reforçado e adicioná-lo a PM_R.

$nCortes = 1$

Inicialização 2:

Considerar a solução em que todos os serviços estão a funcionar em todos os períodos possíveis:

$z_{it} = 0, \forall i \in I^c, t \in TP$

$z_{i2} = 1, \forall i \in I^o$

$z_{it} = 0, \forall i \in I^o, t = 3, ..., T$

$LI = -\infty$

Resolver DSP_z

$LS = V(DSP_z)$

Executar o algoritmo (4.2)

Construir o corte de Benders reforçado e adicioná-lo a PM_R.

$nCortes = 1$

Inicialização 3:

Resolver PM_R sem qualquer corte.

$LI = V(PM_R)$

Resolver DSP_z

$LS = V(DSP_z)$

Executar o algoritmo (4.2)

Construir o corte de Benders reforçado e adicioná-lo a PM_R.

$nCortes = 1$

Apresenta-se de seguida um exemplo de aplicação fortalecendo, em cada iteração, o corte usual.

Exemplo 4.2 *Consideremos os dados da instância do problema de* phase-in/phase-out *utilizado no exemplo 4.1 (página 57) em que se considera:*

$n = 5$, $m = 10$, $I^c = \{1,2\}$, $I^o = \{3,4,5\}$ e $TP = \{1,2,3,4\}$.

Na resolução dos problemas PR_{it}, DSP_z e de PM_R, utilizou-se o general solver IlogCplex 9.0 *[13]. Vamos considerar $\varepsilon = 0.001$. A função objectivo e restrições de PM_R mantêm-se conforme o exemplo 4.1.*

Da mesma forma que no exemplo 4.1, inicializa-se o processo considerando a solução admissível obtida pela heurística apresentada na secção 3.3 do capítulo anterior (Inicialização 1) e que, recordemos, é a seguinte:

	1	2	3	4
1	■	□	□	□
2	■	■	■	■
3	□	□	□	□
4	□	■	■	■
5	□	□	□	□

$z_{11} = z_{42} = 1$

$e\ z_{it} = 0\ para\ i \in I,\ t \in TP,\ (i,t) \neq (1,1)\ e\ (i,t) \neq (4,2)$

custo = 30 080.66

LI = -∞

LS = 30 080.66 (valor obtido pela heurística)

Resolvendo DSP_z, a solução óptima obtida produz os seguinte corte usual:

$$\rho \geq 9214.296 - 1103.568(1 - z_{11}) - 68.864(1 - z_{11} - z_{12}) - 5(1 - z_{11} - z_{12} - z_{13}) - 603.84 - 1351.839(1 - z_{21}) - 149.76(1 - z_{21} - z_{22}) - 1725.12\, z_{32} - 583.1(z_{32} + z_{33}) - 1779.339\, z_{52} - 243.47(z_{52} + z_{53})$$

Mas não será este o corte a adicionar a PM_R. Vamos tentar fortalecê-lo usando o algoritmo 4.2. Este seleciona os pares (i, t) para os quais pode resolvido o problema PR_{it}. Sendo $z_{11} = 1$ ($1 \in I^c$) é resolvido o problema PR_{it} para os pares (1, 2), (1, 3) e (1, 4). Como $3 \in I^o$ e $\sum_{t=2}^{4} z_{3t} = 0$ então PR_{it} pode ser resolvido para os pares (3, 2), (3, 3) e (3, 4). O mesmo sucedendo com $i{=}5 \in I^o$, pois $\sum_{t=2}^{4} z_{5t} = 0$, podendo PR_{it} ser resolvido considerando os pares (5, 2), (5, 3) e (5, 4). Tal como se desejava, a aplicação do algoritmo 4.2 permitiu obter algumas alterações convenientes na solução dual:

v_{it}	1	2	3	4
1	0	0	0.252	0
2	2.516	4.066	0.624	0
3	4.92	2.972	0.112	0
4	4.111	0	0	0
5	3.739	0	0	0

$\pi_{112} = 9$, $\pi_{113} = 6.116$, $\pi_{114} = 5$, $\pi_{122} = 40$, $\pi_{152} = 297$, $\pi_{162} = 70$, $\pi_{182} = 67$, $\pi_{232} = 7.533$, $\pi_{252} = 209.533$, $\pi_{262} = 58.4$, $\pi_{282} = 100.533$, $\pi_{352} = 402.81$, $\pi_{353} = 72.256$, $\pi_{382} = 423.62$, $\pi_{383} = 304.248$, $\pi_{532} = 56$, $\pi_{552} = 475$, $\pi_{553} = 20.312$, $\pi_{582} = 571$, $\pi_{583} = 100.896$ *e para as restantes variáveis π_{ijt} iguais a 0.*

Com esta nova solução constrói-se o primeiro corte a introduzir em PM_R :

$$\begin{aligned}\rho \geq\ & 9214.296 - (0+483)(1-z_{11}) - (0.252 \times 249 + 6.116)(1 - z_{11} - z_{12}) - \\ & (0+5)(1 - z_{11} - z_{12} - z_{13}) - (2.516 \times 240 + 0) - (4.066 \times 240 + 376)(1 - z_{21}) - \\ & (0.624 \times 240 + 0)(1 - z_{21} - z_{22}) - (2.972 \times 192 + 826.43)\, z_{32} - \\ & (0.112 \times 192 + 376.504)(z_{32} + z_{33}) - (0 + 1102)\, z_{52} - (0 + 121.208)(z_{52} + z_{53})\end{aligned}$$

$\Leftrightarrow$

$$\begin{aligned}\rho \geq\ & 9214.296 - 483(1 - z_{11}) - 68.864(1 - z_{11} - z_{12}) - 5(1 - z_{11} - z_{12} - z_{13}) - \\ & 603.84 - 1351.84(1 - z_{21}) - 149.76(1 - z_{21} - z_{22}) - 1397.054\, z_{32} - \\ & 398.008(z_{32} + z_{33}) - 1102\, z_{52} - 121.208(z_{52} + z_{53})\end{aligned}$$

Fazendo uma comparação com o corte usual (antes do fortalecimento - página anterior), repare-se que, para qualquer valor de $\mathbf{z}$*, o membro direito deste último corte é sempre não inferior ao corte usual. Tendo em conta o corte na fase intermédia, é fácil verificar que os coeficientes de* $(1 - z_{11})$*,* z_{32}*,* $(z_{32} + z_{33})$*,* z_{52} *e* $(z_{52} + z_{53})$ *se alteraram (estamos agora a subtrair menor quantidade).*

O corte pode ainda ser simplificado, obtendo-se:

$$\begin{aligned}\rho \geq\ & 6551.992 + 556.864\, z_{11} + 73.864\, z_{12} + 5\, z_{13} + 1501.6\, z_{21} + 149.76\, z_{22} - \\ & 1795.062\, z_{32} - 398.008\, z_{33} - 1223.208\, z_{52} - 121.208\, z_{53}\end{aligned}$$

nCortes = 1

Passo 1:

Depois de introduzir o corte em PM_R, a resolução deste problema produz a seguinte configuração:

	1	*2*	*3*	*4*
1	■	■	■	□
2	■	□	□	□
3	□	□	□	■
4	□	■	■	■
5	□	□	□	□

$z_{13} = z_{21} = z_{34} = z_{42} = 1$

e para os restantes $z_{it} = 0$

$LI = V(PM_R) = 28\ 710.561$

Embora a configuração dada por PM_R seja igual à configuração obtida na primeira iteração do exemplo 4.1, o seu valor, que constitui o primeiro limite inferior produzido, aumentou consideravelmente em relação ao valor produzido com a introdução do primeiro corte usual em PM_R no referido exemplo (passou de 28 089.78 para 28 710.56).

Como se obteve a mesma configuração que no exemplo 4.1 depois de resolvido PM_R, na resolução de DSP_z foi obtida a mesma solução dual.

V(DSP_z)=30 080.66

LS = min { 30 080.66, 29 551.81 } = 29 551.81

Passo 2:

Critério de paragem: $\frac{LS-LI}{LS} = \frac{29551.81-28710.561}{29551.81} = 0.0284 > 0.001$

Passo 3:

A solução de DSP_z origina um corte igual ao segundo corte usual, gerado no referido exemplo e que é o seguinte:

$$\rho \geq 10484.432 - 483(1 - z_{11}) - 62.748(1 - z_{11} - z_{12}) - 11(1 - z_{11} - z_{12} - z_{13}) - 603.84 - 1726.56(1 - z_{21}) - 1159(1 - z_{21} - z_{22}) - 595.2(1 - z_{21} - z_{22} - z_{23}) - 1725.12\, z_{32} - 1139.328(z_{32} + z_{33}) - 435(z_{42} + z_{43} + z_{44}) - 1779.339\, z_{52} - 1106.91(z_{52} + z_{53})$$

Vamos tentar melhorar este corte executando o algoritmo 4.2. Este conduz às seguintes variáveis:

v_{it}	1	2	3	4
1	0	0	0.252	0
2	2.516	4.066	0.624	0
3	4.92	2.973	3.944	0
4	4.111	0	0	1
5	3.739	0	0	0

$\pi_{112} = 9$, $\pi_{114} = 11$, $\pi_{122} = 40$, $\pi_{152} = 297$, $\pi_{162} = 70$, $\pi_{182} = 67$, $\pi_{214} = 6$, $\pi_{223} = 78.84$, $\pi_{224} = 39$, $\pi_{232} = 7.533$, $\pi_{233} = 9$, $\pi_{252} = 151.536$, $\pi_{253} = 257.564$, $\pi_{262} = 58.4$, $\pi_{264} = 62$, $\pi_{274} = 4$, $\pi_{282} = 100.533$, $\pi_{283} = 301.312$, $\pi_{352} = 402.81$, $\pi_{353} = 88.376$, $\pi_{382} = 423.62$, $\pi_{383} = 206.985$, $\pi_{532} = 56$, $\pi_{533} = 83$, $\pi_{552} = 475$, $\pi_{553} = 277.876$, $\pi_{582} = 571$, $\pi_{583} = 402.208$, *com as restantes variáveis π_{ijt} iguais a 0.*

O corte fortalecido a considerar em PM_R é:

$$\rho \geq 10484.432 - (0 + 483)(1 - z_{11}) - (0.252 \times 249 + 0)(1 - z_{11} - z_{12}) - (0 + 11)(1 - z_{11} - z_{12} - z_{13}) - (2.516 \times 240 + 0) - (4.066 \times 240 + 376)(1 - z_{21}) - (0.624 \times 240 + 789.252)(1 - z_{21} - z_{22}) - (0 + 111)(1 - z_{21} - z_{22} - z_{23}) - (2.973 \times 192 + 826.43)\, z_{32} - (3.944 \times 192 + 295.361)(z_{32} + z_{33}) - (1 \times 435 + 0)(z_{42} + z_{43} + z_{44}) - (0 + 1102)\, z_{52} - (0 + 763.084)(z_{52} + z_{53})$$

$\Leftrightarrow$

$$\rho \geq 10484.432 - 483(1 - z_{11}) - 62.748(1 - z_{11} - z_{12}) - 11(1 - z_{11} - z_{12} - z_{13}) - 603.84 - 1351.84(1 - z_{21}) - 939.012(1 - z_{21} - z_{22}) - 111(1 - z_{21} - z_{22} - z_{23}) - 1397.247\, z_{32} - 1052.609(z_{32} + z_{33}) - 435(z_{42} + z_{43} + z_{44}) - 1102\, z_{52} - 763.084(z_{52} + z_{53})$$

$\Leftrightarrow$

$$\rho \geq 6921.992 + 556.748\, z_{11} + 73.748\, z_{12} + 11\, z_{13} + 2401.852\, z_{21} + 1050.012\, z_{22} - 111\, z_{23} - 2449.856\, z_{32} - 1052.609\, z_{33} - 435\, z_{42} - 435\, z_{43} - 435\, z_{44} - 1865.084\, z_{52} - 763.084\, z_{53}$$

nCortes = 2

Passo 1:

Com a introdução do corte em PM_R e resolução deste problema obtém-se a seguinte configuração:

	1	2	3	4
1	■	□	□	□
2	■	■	■	□
3	□	□	□	■
4	□	■	■	■
5	□	□	□	□

$z_{11} = z_{23} = z_{34} = z_{42} = 1$
e para os restantes $z_{it} = 0$

$LI = V(PM_R) = 28\ 967.549$

A resolução de DSP_z permite obter:

V(DSP_z)=28 973.665

LS = min { 29 551.81, 28 973.665 } = 28973.665

Passo 2:

Critério de paragem: $\frac{LS-LI}{LS} = \frac{28973.665-28967.549}{28973.665} = 0.0002 < 0.001$

Passo 4:

Stop. A configuração associada a LS é óptima para o problema original. $\diamondsuit$

4.3.2 Experiência computacional

Nesta subsecção apresentam-se os resultados computacionais obtidos com a aplicação da decomposição de Benders usando o algoritmo (4.2) para fortalecimento do corte usual. A tabela 4.7 apresenta os resultados obtidos para o grupo de instâncias com 10 localizações e as tabelas 4.8 e 4.9 para as instâncias que consideram, respectivamente, 20 e 50 localizações. Para cada instância apresenta-se o tempo de *CPU* (em segundos) assim como o número de cortes gerados.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		3	0.62	1	0.56	1	0.56	1
5		5	1.38	3	1.30	3	1.17	3
		8	1.11	3	1.05	3	0.77	2
		3	3.84	3	3.72	3	3.81	3
10	20	5	3.86	4	3.70	4	3.78	4
		8	2.02	2	1.77	2	1.86	2
		3	10.16	4	9.81	4	28.50	6
15		5	594.41	8	1130.83	7	1129.86	8
		8	6.38	2	3.02	2	2.97	2
		3	1.48	1	0.66	1	0.69	1
5		5	2.49	3	1.53	3	1.09	2
		8	2.81	4	1.81	4	2.00	5
		3	6.62	2	2.84	2	2.89	2
10	50	5	5.70	2	2.10	2	2.22	2
		8	12.46	6	8.67	6	9.70	7
		3	60.49	7	53.85	7	27.71	7
15		5	27.67	3	15.39	3	18.89	4
		8	136.52	7	124.36	7	90.89	10
		3	4.11	2	1.81	2	1.61	2
5		5	3.48	1	0.75	1	1.42	2
		8	3.59	1	0.70	1	0.69	1
		3	14.72	3	4.72	3	5.33	3
10	100	5	11.39	2	3.31	2	3.33	2
		8	10.98	1	1.95	1	1.95	1
		3	34.86	3	9.20	3	10.76	4
15		5	45.57	4	24.13	4	34.87	5
		8	45.02	3	9.55	3	9.09	3

Tabela 4.7: Tempo de *CPU* (seg) *#cortes* - 10 localizações.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	#J	#I^c	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		8	10.69	1	1.12	1	1.19	1
5		10	8.33	1	1.11	1	1.09	1
		16	23.65	13	13.13	13	13.01	13
		8	85.92	6	49.52	6	48.64	7
10	50	10	200.19	10	151.03	10	268.56	13
		16	129.22	7	73.97	7	50.23	8
		8	1369.04	11	5586.60	10	5170.84	11
15		10	4476.26	16	2072.72	20	2900.77	17
		16	1157.47	12	1612.20	10	1522.75	10
		8	35.91	4	5.20	4	5.22	4
5		10	28.89	2	2.61	2	2.64	2
		16	33.44	8	10.95	8	13.92	10
		8	129.16	5	20.42	5	23.70	6
10	100	10	169.61	8	30.22	8	31.84	8
		16	150.97	8	41.95	7	61.69	9
		8	474.23	5	154.99	5	110.16	6
15		10	24330.20	11	>12h	-	33364.63	10
		16	679.38	6	477.49	6	109.31	5

Tabela 4.8: Tempo de *CPU* (seg) e #*cortes* - 20 localizações.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	#J	#I^c	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		20	92.48	2	8.56	2	8.70	2
5		25	105.30	3	12.48	3	16.08	4
		45	161.22	14	59.56	15	52.03	13
		20	1165.72	14	3550.53	16	3037.21	15
10	100	25	4018.16	12	2130.86	12	2290.62	12
		45	785.37	11	135.05	10	196.11	15
		20	6130.49	9	16.31	1	37.77	2
15		25	*out mem.*	-	7.86	1	*out mem.*	-
		45	1405.00	7	475.65	8	970.26	9

Tabela 4.9: Tempo de *CPU* (seg) e #*cortes* - 50 localizações.

⌜ *Para a instância assinalada com 'out mem.' na tabela 4.9, o solver apresentou um erro de memória aquando da resolução de* PM_R. ⌟

Façamos agora uma breve análise dos resultados obtidos com a introdução de cortes melhorados no problema mestre relaxado, à semelhança do que foi feito com os resultados da secção anterior.

variante	tempo	cortes
Inicialização 1	39.03	3.1
Inicialização 2	52.71	3.1
Inicialização 3	51.79	3.5

Tabela 4.10: Valores médios - 10 localizações.

Observando a tabela 4.7, com excepção da instância com 20 clientes/15 períodos/$\#I^c$=5, a segunda variante de inicialização domina a forma de inicialização 1 no que diz respeito ao tempo de *CPU*. Em relação ao número de cortes gerados, é curioso verificar que, neste grupo de instâncias, a forma de inicialização 2 gera igual número de cortes que a inicialização 1 em 26 das 27 instâncias e exactamente na instância acima referida na qual gastou mais tempo, gera menos um corte. O que é interessante, uma vez que, quando se considera a inicialização 2 no algoritmo 4.3, o primeiro corte a ser adicionado a PM_R, na prática, não pode ser fortalecido, ou seja, na aplicação do algoritmo 4.2, não existe nenhum par (i, t) para o qual possa ser resolvido o problema PR_{it}. Relativamente à inicialização 3, pode dizer-se que para este grupo de instâncias, o desempenho é semelhante ao das restantes formas de inicialização.

Comparando estes resultados com a tabela 4.4, é notável o decréscimo conseguido nos tempos de *CPU*. Veja-se que com a introdução de um corte potencialmente mais forte, o tempo médio gasto usando cada uma das variantes de inicialização, é reduzido para menos de 10 % do tempo médio obtido com o uso do corte usual e o número médio de cortes gerados reduziu mais de 60 %.

Para o grupo de instâncias com 20 localizações, os valores médios obtidos apresentam-se na tabela 4.8. Desta informação está excluída a instância com 100 clientes/15 períodos/$\#I^c$=10 por não se ter obtido uma solução óptima em tempo inferior a 12 horas quando considerada a variante 2.

variante	tempo	cortes
Inicialização 1	538.96	7.2
Inicialização 2	606.19	7.2
Inicialização 3	607.97	7.7

Tabela 4.11: Valores médios - 20 localizações.

A variante de inicialização 2 tem sido até ao momento a que, em geral, melhores resultados tem produzido. No entanto, neste grupo de instâncias surgiram algumas dificuldades, especialmente em 3 instâncias: com 50 clientes/15 períodos/$\#I^c$=8, 50 clientes/15 períodos/$\#I^c$=16 e também com 100 clientes/15 períodos/$\#I^c$=10. Esta última ultrapassou os 43 200 segundos (12 horas). Mais uma vez, a dificuldade sentiu-se em instâncias nas quais se considera o horizonte de planeamento particionado em 15 períodos de tempo.

De acordo com a tabela 4.11, quer o tempo médio de *CPU* quer o número médio de cortes são bastante equilibrados entre as três variantes de inicialização.

De notar que as duas instâncias deste grupo que não se conseguiram resolver em menos de 12 horas usando o corte usual, foram agora resolvidas em tempo útil (apenas a variante 2 ultrapassou as 12 horas). Comparando os resultados obtidos para este grupo com a tabela 4.2, são evidentes os significativos decréscimos quer ao nível dos tempos de *CPU* quer ao nível do número de cortes.

A tabela que se segue diz respeito aos valores médios das instâncias com 50 localizações não considerando as instâncias para as quais ocorreu um erro de memória na resolução (pelo *general solver*) do problema mestre relaxado.

variante	tempo	cortes
Inicialização 1	1732.97	9.0
Inicialização 2	798.63	8.4
Inicialização 3	826.09	9.0

Tabela 4.12: Valores médios - 50 localizações.

Comparando com os resultados médios obtidos usando o corte usual (tabela 4.6 - página 69), verificam-se melhorias muito acentuadas nos tempos de *CPU* e no número médio de cortes gerados para as instâncias de maior dimensão. Para as instâncias que consideram 5 períodos no horizonte de planeamento, os tempos não são muito diferentes mas o número de cortes gerados diminuiu.

A segunda variante de inicialização é, mais uma vez, a que conduz a tempos de execução inferiores, nomeadamente em relação à primeira variante. Embora a diferença não seja muito significativa, o número médio de cortes gerados pela variante 2 é inferior ao número médio de cortes gerados quando são usadas as outras variantes de inicialização.

Começa a ser questionável a utilização de uma solução admissível obtida pela heurística na inicialização deste processo. À medida que aumenta a dimensão das instâncias, os tempos para obtenção de uma solução admissível também aumenta (tal como vimos nos resultados apresentados no capítulo 3 - página 41).

Em síntese, o reforço do corte permitiu, de facto, reduções consideráveis no número de cortes gerados e, consequentemente nos tempos de *CPU* produzidos.

4.4 Um corte obtido heuristicamente a partir do corte usual

Na secção anterior geraram-se cortes mais fortes, tendo estes sido reforçados utilizando um problema de programação linear, resolvido utilizando o *general solver IlogCplex 9.0* [13]. Os resultados melhoraram de forma significativa relativamente ao que se tinha obtido usando o corte usual. No entanto, em cada iteração é necessário invocar um procedimento (algoritmo 4.2) no qual é necessário resolver um problema de programação linear para cada par (i, t) selecionado. Ora se o número de pares (i, t) for elevado, mesmo tratando-se de um problema de programação linear, tal pode tornar-se dispendioso em termos de tempo de *CPU*, embora já tenhamos visto que o número de cortes gerados para obter a solução óptima diminuiu. O que se propõe nesta secção é que o fortalecimento do corte usual (secção 4.2) se faça por meio de uma heurística que usa na sua estratégia informação do dual de SP_z sem se considerar as restrições de afectação (4.3) e (4.4).

4.4.1 Metodologia

Considere-se de novo o problema PR_{it} (página 72) para um certo par $(i,\, t)$, $i \in I$ e $t \in TP$.

Como $\pi_{ijt} \geq 0$ e $\pi_{ijt} \geq \lambda_{jt} - c_{ijt} - d_{jt}\, v_{it}$, $\forall j \in J$, então $\pi_{ijt} \geq \max\{0, \lambda_{jt} - c_{ijt} - d_{jt}\, v_{it}\}$. Sendo o problema de minimização e tendo as variáveis π coeficiente positivo na função objectivo, pode fixar-se π_{ijt}, $j \in J$, no seu valor mais baixo, ou seja,

$$\pi_{ijt} = \max\{0, \lambda_{jt} - c_{ijt} - d_{jt}\, v_{it}\}$$

Esta igualdade permite reescrever o problema PR_{it} na seguinte forma condensada:

$$\min_{v_{it} \geq 0} \quad v_{it}\, Q_i \; + \sum_{j \in J} \max\{0, \lambda_{jt} - c_{ijt} - d_{jt}\, v_{it}\} \tag{4.26}$$

Para resolver este problema propõe-se aqui um procedimento heurístico seguindo uma ideia apresentada por Van Roy [49], que parte de uma solução do dual de SP_z sem considerar neste último problema os conjuntos de restrições que estão associados às variáveis π_{ijt}, atribuindo-lhes, consequentemente, valor zero. Tenta-se então diminuir o valor das variáveis v_{it} à custa de aumentos nos valores das variáveis π_{ijt}.

Repare-se que, em (4.26), se se diminuir Δ unidades em v_{it}, então $v_{it}\, Q_i$ diminui $\Delta\, Q_i$ unidades e $\sum_{j \in J} \{\lambda_{jt} - c_{ijt} - d_{jt}\, v_{it}\}$ aumenta $\Delta \sum_{j \in J} d_{jt}$ unidades. Assim, se a diminuição ocorrida na primeira parcela compensar o aumento ocorrido na segunda parte, isto é, se $Q_i > \sum_{j \in J} d_{jt}$, então vale a pena decrementar v_{it} em Δ unidades.

E que quantidade Δ deve ser considerada?

Note-se que, se se considerarem as variáveis $\pi_{ijt} = 0$, v_{it} é admissível se e só se $\lambda_{jt} - d_{jt}\, v_{it} \leq c_{ijt}$. Como $v_{it} \geq 0$, então na solução óptima teremos $v_{it} = \max\{0, \max_{j \in J} \dfrac{\lambda_{jt} - c_{ijt}}{d_{jt}}\}$.

Se $v_{it} = 0$, nada se pode decrementar pois tem-se sempre $v_{it} \geq 0$. Se $v_{it} \neq 0$, então necessariamente tem-se

$$v_{it} = \max_{j \in J} \frac{\lambda_{jt} - c_{ijt}}{d_{jt}} \tag{4.27}$$

Tem-se:

$$\begin{aligned} v_{it}\, Q_i \; + \sum_{j \in J} \max\{0, \lambda_{jt} - c_{ijt} - d_{jt}\, v_{it}\} = v_{it}\, Q_i \; + \; & \max\{0, \lambda_{1t} - c_{i1t} - d_{1t}\, v_{it}\} + \\ & \max\{0, \lambda_{2t} - c_{i2t} - d_{2t}\, v_{it}\} + \\ & \max\{0, \lambda_{3t} - c_{i3t} - d_{3t}\, v_{it}\} + \\ & \max\{0, \lambda_{4t} - c_{i4t} - d_{4t}\, v_{it}\} + \\ & \vdots \\ & \max\{0, \lambda_{mt} - c_{imt} - d_{mt}\, v_{it}\} \end{aligned}$$

Suponhamos, sem perda de generalidade, que $\frac{\lambda_{4t} - c_{i4t}}{d_{4t}} > \frac{\lambda_{3t} - c_{i3t}}{d_{3t}} > \frac{\lambda_{2t} - c_{i2t}}{d_{2t}} > 0$ e $\frac{\lambda_{1t} - c_{i1t}}{d_{1t}} < 0$. Suponhamos ainda que $\frac{\lambda_{5t} - c_{i5t}}{d_{5t}}, \ldots, \frac{\lambda_{mt} - c_{imt}}{d_{mt}} < 0$.

Assim, de acordo com (4.27) será $v_{it} = \frac{\lambda_{4t} - c_{i4t}}{d_{4t}} \quad \Rightarrow \quad \lambda_{4t} - c_{i4t} - d_{4t}\, v_{it} = 0$ e que

$$\begin{aligned} \lambda_{3t} - c_{i3t} - d_{3t}\, v_{it} &< 0 \\ \lambda_{2t} - c_{i2t} - d_{2t}\, v_{it} &< 0 \\ \lambda_{1t} - c_{i1t} - d_{1t}\, v_{it} &< 0 \end{aligned}$$

Fazendo v_{it} decrescer, quando alcançar o valor $\frac{\lambda_{3t} - c_{i3t}}{d_{3t}}$ terá ocorrido uma diminuição de $\left(\frac{\lambda_{4t} - c_{i4t}}{d_{4t}} - \frac{\lambda_{3t} - c_{i3t}}{d_{3t}}\right)$ unidades no valor desta variável, o que faz com que:

$$\begin{aligned} \lambda_{4t} - c_{i4t} - d_{4t}\, v_{it} &> 0 \\ \lambda_{3t} - c_{i3t} - d_{3t}\, v_{it} &= 0 \\ \lambda_{2t} - c_{i2t} - d_{2t}\, v_{it} &< 0 \\ \lambda_{1t} - c_{i1t} - d_{1t}\, v_{it} &< 0 \end{aligned}$$

Teremos, então, $v_{it} = \frac{\lambda_{3t} - c_{i3t}}{d_{3t}}$, $\pi_{i4t} > 0$ e $\pi_{i3t} = \pi_{i2t} = \pi_{i1t} = 0$. Mas a diminuição só faz sentido se $Q_i > d_{4t}$. Suponhamos que esta desigualdade se verifica. Quando se considera nova diminuição de v_{it}, pode tentar-se diminuir $\left(\frac{\lambda_{3t} - c_{i3t}}{d_{3t}} - \frac{\lambda_{2t} - c_{i2t}}{d_{2t}}\right)$ unidades, o que conduzirá a

$$\begin{aligned} \lambda_{4t} - c_{i4t} - d_{4t}\, v_{it} &> 0 \\ \lambda_{3t} - c_{i3t} - d_{3t}\, v_{it} &> 0 \\ \lambda_{2t} - c_{i2t} - d_{2t}\, v_{it} &= 0 \end{aligned}$$

$$\lambda_{1t} - c_{i1t} - d_{1t}\, v_{it} < 0$$

Pelo que, se $Q_i > d_{4t} + d_{3t}$ então é vantajoso o decréscimo em v_{it}. Suponhamos que assim é. Repare-se agora na próxima tentativa para diminuir v_{it} (que neste momento tem valor $\frac{\lambda_{2t}-c_{i2t}}{d_{2t}}$). O próximo quociente a ser considerado é o que está associado a $j = 1$, que se considerou ser negativo. Naturalmente que não se pode diminuir a variável v_{it} até um valor negativo, mas pode diminuir-se a variável até que atinja o valor zero. Nesta situação, ter-se-ia:

$$\begin{aligned}
\lambda_{4t} - c_{i4t} - d_{4t}\, v_{it} &> 0\\
\lambda_{3t} - c_{i3t} - d_{3t}\, v_{it} &> 0\\
\lambda_{2t} - c_{i2t} - d_{2t}\, v_{it} &> 0\\
\lambda_{1t} - c_{i1t} - d_{1t}\, v_{it} &< 0
\end{aligned}$$

Repare-se que π_{i1t} será sempre nula pois mesmo havendo diminuição de v_{it}, $\lambda_{1t} - c_{i1t} - d_{1t}\, v_{it}$ será sempre negativo qualquer que seja a quantidade decrementada. O mesmo sucedendo com os restantes quocientes negativos, o que fará com que as variáveis π_{ijt} associadas tenham valor zero.

A quantidade Δ será assim definida através da diferença entre os sucessivos quocientes positivos dispostos por ordem decrescente. Quando se encontra o primeiro quociente negativo, Δ tomará o valor que v_{it} tiver nessa altura.

O procedimento heurístico que conduzirá à eventual alteração do valor da variável v_{it} e, consequentemente, de variáveis π_{ijt}, $j \in J$, pode ser descrito da seguinte forma para um dado par (i, t):

Algorithm 4.4 Procedimento para alteração do valor de v_{it} e π_{ijt}, $j \in J$

$J_{it} = \emptyset$, $\Delta = 0$, $\pi_{ijt} = 0$, $\forall j \in J$
Calcular $k_j = \frac{\lambda_{jt}-c_{ijt}}{d_{jt}}$, $\forall j \in J$
while $v_{it} > 0 \wedge \Delta \neq 0$ **do**
  $j^* = \arg\{\max k_j\colon j \notin J_{it}\}$
  $J_{it} = J_{it} \cup \{j^*\} \cup \{j \in J\colon j \notin J_{it} \wedge k_j = k_{j^*}\}$
  $k_{j'} = \max k_j\colon j \notin J_{it}$
  if $Q_i > \sum_{j \in J_{it}} d_{jt}$ **then**
    if $k_{j'} < 0$ **then**
      $\Delta = k_{j^*}$
      $\pi_{ijt} = \pi_{ijt} + \Delta\, d_{jt}$, $\forall j \in J_{it}$
      $v_{it} = 0$
    if $k_{j'} \geq 0$ **then**
      $\Delta = k_{j^*}$ - $k_{j'}$
      $\pi_{ijt} = \pi_{ijt} + \Delta\, d_{jt}$, $\forall j \in J_{it}$
      $v_{it} = v_{it}$ - Δ
  else
    Stop

Este reforço pode ser feito para diversos pares (i, t) que satisfaçam as condições referidas na subsecção 4.3.1 (página 71). O algoritmo 4.5 sintetiza isso mesmo. São identificados os pares (i, t) com os quais se procede a uma tentativa de reforço invocando o algoritmo 4.4 como subrotina. Como o reforço tenta tirar proveito da diminuição das variáveis v_{it} à custa do aumento de alguma ou algumas variáveis π_{ijt}, $j \in J$ se, após a resolução do problema dual de SP_z, para algum par (i, t) v_{it} tiver valor nulo então nada se pode diminuir no valor desta variável, pelo que não se considera a execução do algoritmo 4.4 para esse par (i, t).

Algorithm 4.5 Procedimento de reforço

Resolver o dual de SP_z sem os conjuntos de restrições (4.3) e (4.4).
for $i \in I^c$ **do**
  Se existir τ tal que $z_{i\tau} = 1$
  for t $= \tau + 1, \ldots, T$ **do**
    if $v_{it} > 0$ **then**
      Executar o algoritmo 4.4 para o par (i, t)
for $i \in I^o$ **do**
  if $\sum_{\tau=2}^{T} z_{i\tau} = 0$ **then**
    for $t = 2, \ldots, T$ **do**
      if $v_{it} > 0$ **then**
        Executar o algoritmo 4.4 para o par (i, t)
  else
    Determinar τ tal que $z_{i\tau} = 1$
    for $t = 2, \ldots, \tau - 1$ **do**
      if $v_{it} > 0$ **then**
        Executar o algoritmo 4.4 para o par (i, t)

O algoritmo da decomposição de Benders com a introdução de cortes reforçados utilizando a metodologia acabada de descrever é formalizado no algoritmo 4.6 sendo, em seguida, efectuada a alteração em cada uma das variantes de inicialização em conformidade com o algoritmo 4.6. Neste algoritmo, ε representa uma quantidade arbitrariamente pequena.

Algorithm 4.6 Algoritmo da decomposição de Benders usando uma heurística de reforço

Inicialização 1, 2 ou 3
Passo 1:
Resolver PM_R
$LI = V(PM_R)$
Actualizar vector $\mathbf{z}$
Resolver DSP_z para obter (λ, π, v)
$LS = \min\{LS, V(DSP_z)\}$
Passo 2:
se $\frac{LS-LI}{LS} < \varepsilon$ ir para o **Passo 4**
Passo 3:
Executar o algoritmo 4.5
Construir o corte de Benders reforçado e adicioná-lo a PM_R
$nCortes = nCortes + 1$
Voltar ao **Passo 1**
Passo 4:
Stop. A solução associada ao valor de LS é a solução óptima.

Inicialização 1:

Obter uma solução admissível usando a heurística apresentada na secção 3.3 e seja V_H o seu valor.

$LI = -\infty$

$LS = V_H$

Resolver DSP_z

Executar o algoritmo 4.5

Construir o de Benders reforçado e adicioná-lo a PM_R.

$nCortes = 1$

Inicialização 2:

Considerar a solução em que todos os serviços estão a funcionar em todos os períodos possíveis:

$z_{it} = 0, \forall i \in I^c, t \in TP$

$z_{i2} = 1, \forall i \in I^o$

$z_{it} = 0, \forall i \in I^o, t = 3, ..., T$

$LI = -\infty$

Resolver DSP_z

$LS = V(DSP_z)$

Executar o algoritmo 4.5

Construir o de Benders reforçado e adicioná-lo a PM_R.

$nCortes = 1$

Inicialização 3:

Resolver o PM_R sem qualquer corte.

$LI = V(PM_R)$

Resolver DSP_z

$LS = V(DSP_z)$

Executar o algoritmo 4.5

Construir o de Benders reforçado e adicioná-lo a PM_R.

$nCortes = 1$

Vejamos um pequeno exemplo ilustrativo da aplicação do método da decomposição de Benders introduzindo em cada iteração um corte reforçado usando a heurística sistematizada no algoritmo 4.5.

Exemplo 4.3 *Considere-se novamente a instância apresentada no exemplo 4.1 (página 57). Considere-se, ainda, a variante de inicialização 1 e* ε*=0.001.*

n = *10,* m = *10,* $I^c = \{1, 2\}$*,* $I^o = \{3, 4, 5\}$ *e* $TP = \{1, 2, 3, 4\}$*.*

A configuração admissível inicial é:

	1	2	3	4
1	■	□	□	□
2	■	■	■	■
3	□	□	□	□
4	□	■	■	■
5	□	□	□	□

$z_{11} = z_{42} = 1$

e z_{it} = *0 para os retantes*

Valor = 30 080.66

LI = -∞

LS = 30 080.66 (valor fornecido pela heurística)

Resolvendo DSP_z*, foi obtida a seguinte solução:*

λ_{jt}	1	2	3	4
1	4	13	10.872	9
2	108	149	46.72	26
3	208.096	266	132.464	11
4	787	455	472	25
5	701.54	997	513.312	56
6	86.87	185	186	32
7	113	62	66	64
8	864.516	1102	668.89	64
9	99	62	61	58
10	233	71	70	75

v_{it}	1	2	3	4
1	0	4.432	0.252	0
2	2.516	4.066	0.624	0
3	4.92	8.985	3.037	0
4	4.111	0	0	0
5	3.739	7.089	0.97	0

$$\sum_{j \in J} \sum_{t \in TP} \lambda_{jt} = 9214.296$$

$\pi_{113} = 6.116$, $\pi_{114} = 5$, $\pi_{232} = 7.533$, $\pi_{252} = 209.533$, $\pi_{262} = 58.4$, $\pi_{282} = 100.533$

Na execução do algoritmo 4.5, começa-se por resolver o dual de SP_z *não considerando as restrições de afectação. Tal conduz à seguinte solução:*

λ_{jt}	1	2	3	4
1	4	13	10.872	9
2	108	149	46.72	26
3	208.096	258.466	132.464	11
4	787	455	472	25
5	701.54	787.466	513.312	56
6	86.87	126.6	186	32
7	113	62	66	64
8	864.516	1001.466	668.89	64
9	99	62	61	58
10	233	71	70	75

v_{it}	1	2	3	4
1	0	3	2.29	1.666
2	2.516	4.066	0.624	0
3	4.92	6.3	3.037	0
4	4.111	0	0	0
5	3.739	4.85	0.97	0

z_{11} = 1 e $1 \in I^c \Rightarrow$ executar o algoritmo 4.4 para os pares (i, t)= (1, 2), (1, 3) e (1, 4).

A aplicação do algoritmo 4.4 vai ser exemplificada apenas para estes 3 pares, o que não será feito em detalhe para os que serão selecionados ao longo do que resta do exemplo, apresentando-se apenas o resultado de cada aplicação.

Para o par (i, t) = (1, 2) :

$J_{12} = \emptyset$, $\Delta = 0$, $\pi_{1j2} = 0$, $\forall\, j \in J$

j	1	2	3	4	5	6	7	8	9	10
k_j	3	1.333	-0.3927	-4.482	1.305	0.4833	-1.909	-0.3457	-4	-2.905

$v_{12} = 3 > 0$
$j^* = 1$ e $k_1 = 3 \Rightarrow J_{12} = \{1\}$
$k_{j'} = 1.333 \geq 0$
$249 > d_{12} = 3$
$\Delta = 3 - 1.333 = 1.667$
$\pi_{112} = 0 + 1.667 \times 3 = 5$
$v_{12} = 3 - 1.667 = 1.333 > 0$ e $\Delta = 1.667 \neq 0$

$j^* = 2$ e $k_2 = 1.333 \Rightarrow J_{12} = \{1,2\}$
$k_{j'} = 1.305 \geq 0$
$249 > d_{12} + d_{22} = 3 + 30 = 33$
$\Delta = 1.333 - 1.305 = 0.02786$
$\pi_{112} = 5 + 0.02786 \times 3 = 5.0835$
$\pi_{122} = 0 + 0.02786 \times 30 = 0.8358$
$v_{12} = 1.33(3) - 0.02786 = 1.305 > 0$ e $\Delta = 0.02786 \neq 0$

$j^* = 5$ e $k_5 = 1.305 \Rightarrow J_{12} = \{1,2,5\}$
$k_{j'} = 0.483(3) \geq 0$
$249 > d_{12} + d_{22} + d_{52} = 3 + 30 + 67 = 100$
$\Delta = 1.305 - 0.483(3) = 0.822$

$\pi_{112} = 5.0835 + 0.822 \times 3 = 7.55$
$\pi_{122} = 0.8358 + 0.822 \times 30 = 25.5$
$\pi_{152} = 0 + 0.822 \times 67 = 55.083(3)$
$v_{12} = 0.483(3) > 0$ e $\Delta = 0.822 \neq 0$

$j^* = 6$ e $k_6 = 0.483(3) \Rightarrow J_{12} = \{1, 2, 5, 6\}$
$k_{j'} = -0.3457 < 0$
$249 > d_{12} + d_{22} + d_{52} + d_{62} = 3 + 30 + 67 + 24 = 124$
$\Delta = 0.483(3)$
$\pi_{112} = 7.55 + 0.483(3) \times 3 = 9$
$\pi_{122} = 25.5 + 0.483(3) \times 30 = 40$
$\pi_{152} = 55.083(3) + 0.483(3) \times 67 = 87.466$
$\pi_{162} = 0 + 0.483(3) \times 24 = 11.6$
$v_{12} = 0 \not> 0 \Rightarrow Stop$

Para o par (i, t) = (1, 3) :

$J_{13} = \emptyset$, $\Delta = 0$, $\pi_{1j3} = 0$, $\forall\, j \in J$

j	1	2	3	4	5	6	7	8	9	10
k_j	2.29	-2.376	-4.54	-5.09	-3.836	0.252	-1.95	-2.645	-3.55	-2.796

$v_{13} = 2.29 > 0$
$j^* = 1$ e $k_1 = 2.29 \Rightarrow J_{13} = \{1\}$
$k_{j'} = 0.25 \geq 0$
$249 > d_{13} = 3$
$\Delta = 2.29 - 0.252 = 2.038$
$\pi_{113} = 0 + 2.038 \times 3 = 6.116$
$v_{13} = 2.29 - 2.038 = 0.252 > 0$ e $\Delta = 2.038 \neq 0$

$j^* = 6$ e $k_6 = 0.252 \Rightarrow J_{13} = \{1, 6\}$
$k_{j'} = -\ 1.954 < 0$
$249 \not> d_{13} + d_{63} = 250$

Passando ao próximo par, (i, t) = (1, 4) :

$J_{14} = \emptyset$, $\Delta = 0$, $\pi_{1j4} = 0$, $\forall\, j \in J$

j	1	2	3	4	5	6	7	8	9	10
k_j	1.666	-2.812	-7.692	-11.458	-10.953	-3.2	-2.14	-7.972	-2.9	-2.679

$v_{14} = 1.666 > 0$

$j^* = 1$ e $k_1 = 1.666 \Rightarrow J_{14} = \{1\}$
$k_{j'} = -2.14 < 0$
$249 > d_{14} = 3$
$\Delta = 1.666$
$\pi_{114} = 0 + 1.666 \times 3 = 5$
$v_{14} = 0$

Como $\sum_{t=2}^{4} z_{3t} = 0$ *e* $3 \in I^o \Rightarrow$ *executar o algoritmo (4.4) para os pares (i, t)= (3, 2), (3, 3) e (3, 4).*

Para o par (i, t) = (3, 2), após a execução do algoritmo 4.4 obteve-se: $v_{32} = 2.769$, $\pi_{352} = 206.918$ *e* $\pi_{382} = 342.837$.

Para o par (i, t) = (3, 3), após a execução do algoritmo 4.4 obteve-se: $v_{33} = 0.112$, $\pi_{353} = 72.256$ *e* $\pi_{383} = 304.248$.

Para o par (i, t) = (3, 4), como $v_{34} = 0$ *nada se pode diminuir.*

Como $\sum_{t=2}^{4} z_{5t} = 0$ *e* $5 \in I^o \Rightarrow$ *executar o algoritmo 4.4 para os pares (i, t)= (5, 2), (5, 3) e (5, 4).*

Para o par (i, t) = (5, 2), após a execução do algoritmo 4.4 tem-se: $v_{52} = 0$, $\pi_{532} = 48.466$, $\pi_{552} = 265.466$ *e* $\pi_{582} = 470.466$.

Para o par (i, t) = (5, 3), após a execução do algoritmo 4.4 obteve-se: $v_{53} = 0$, $\pi_{553} = 20.312$ *e* $\pi_{583} = 100.896$.

Para o par (i, t) = (5, 4), como $v_{54} = 0$ *nada se pode diminuir.*

Em resumo, a informação dual para a construção do primeiro corte a introduzir em PM_R *é a seguinte:*

v_{it}	1	2	3	4
1	0	0	0.252	0
2	2.516	4.066	0.624	0
3	4.92	2.769	0.112	0
4	4.111	0	0	0
5	3.739	0	0	0

$\sum_{j \in J} \sum_{t \in TP} \lambda_{jt} = 9214.296$

$\pi_{112} = 9$, $\pi_{113} = 6.116$, $\pi_{114} = 5$, $\pi_{122} = 40$, $\pi_{152} = 87.466$, $\pi_{162} = 11.6$, $\pi_{232} = 7.533$,

π_{252} = *209.53*, π_{262} = *58.4*, π_{282} = *100.533*, π_{352} = *206.918*, π_{353} = *72.256*, π_{382} = *342.837*, π_{383} = *304.248*, π_{532} = *48.466*, π_{552} = *265.466*, π_{553} = *20.312*, π_{582} = *470.466*, π_{583} = *100.896 e os restantes* π_{ijt} *iguais a 0.*

$$\begin{aligned}\rho \geq\ & 9214.296 - (0 + 148.066)(1 - z_{11}) - (0.252 \times 249 + 6.116)(1 - z_{11} - z_{12}) - \\ & (0 + 5)(1 - z_{11} - z_{12} - z_{13}) - (2.516 \times 240 + 0) - (4.066 \times 240 + 375.996)(1 - z_{21}) - \\ & (0.624 \times 240 + 0)(1 - z_{21} - z_{22}) - (2.769 \times 192 + 549.755)\, z_{32} - \\ & (0.112 \times 192 + 376.504)(z_{32} + z_{33}) - (0 + 784.398)\, z_{52} - (0 + 121.208)(z_{52} + z_{53})\end{aligned}$$

$\Leftrightarrow$

$$\begin{aligned}\rho \geq\ & 9214.296 - 148.06(1 - z_{11}) - 68.864(1 - z_{11} - z_{12}) - 5(1 - z_{11} - z_{12} - z_{13}) - \\ & 603.84 - 1351.836(1 - z_{21}) - 149.76(1 - z_{21} - z_{22}) - 1081.403\, z_{32} - 398.008(z_{32} + z_{33}) - \\ & 784.398\, z_{52} - 121.208(z_{52} + z_{53})\end{aligned}$$

$\Leftrightarrow$

$$\begin{aligned}\rho \geq\ & 6886.936 + 221.924\, z_{11} + 73.864\, z_{12} + 5\, z_{13} + 1501.836\, z_{21} + - \\ & 149.76\, z_{22} - 1479.411\, z_{32} - 398.008\, z_{33} - 905.606\, z_{52} - 121.208\, z_{53}\end{aligned}$$

nCortes = 1

Passo 1:

Depois de introduzido em PM_R *o corte acima, resolve-se este último problema sendo a configuração obtida a seguinte:*

	1	*2*	*3*	*4*
1	■	□	□	□
2	■	■	■	□
3	□	□	□	■
4	□	■	■	■
5	□	□	□	□

$z_{11} = z_{23} = z_{34} = z_{42} = 1$

e para os restantes $z_{it} = 0$

LI = V(PM_R*) = 28 921.665*

De realçar o valor obtido para o limite inferior. No exemplo 4.1, página 57, a resolução de PM_R *conduziu a um limite inferior de 28 089.785. Veja-se o impacto da introdução do corte reforçado usando este procedimento heurístico. Neste caso o limite inferior obtido é de 28 921.665, um valor muito próximo do valor óptimo do problema.*

Resolvendo DSP_z *com a configuração obtida no problema mestre relaxado, obtém-se a seguinte solução dual:*

V(DSP$_z$) = 28 973.665

LS = min { 30 080.66, 28 973.665 } = 28 973.665

$\sum_{j \in J} \sum_{t \in TP} \lambda_{jt}$ *= 9 701.296*

v_{it}	1	2	3	4
1	0	4.432	0.252	0
2	2.516	4.066	0.624	2.48
3	4.92	8.985	3.037	0
4	4.111	0	0	1
5	3.739	7.089	0.97	0

$\pi_{113} = 6.116$, $\pi_{114} = 11$, $\pi_{232} = 7.533$, $\pi_{252} = 209.533$, $\pi_{262} = 58.4$, $\pi_{282} = 100.533$ *e para os restantes* $\pi_{ijt} = 0$.

Passo 2:

Critério de paragem: $\frac{LS-LI}{LS} = \frac{28973.665-28921.665}{28973.665} = 0.00179 > 0.001$

Passo 3:

Na sequência da aplicação do algoritmo 4.5, resolve-se o dual do problema SP_z sem considerar as restrições de afectação, obtendo-se a solução:

λ_{jt}	1	2	3	4
1	4	13	10.872	15
2	108	149	46.72	65
3	208.096	258.466	132.464	50
4	787	455	472	110
5	701.54	787.466	513.312	120
6	86.87	126.6	186	94
7	113	62	66	85
8	864.516	1001.466	668.896	172
9	99	62	61	68
10	233	71	70	128

v_{it}	1	2	3	4
1	0	3	2.29	3.666
2	2.516	4.066	0.624	2.48
3	4.92	6.303	3.037	0
4	4.111	0	0	1
5	3.739	4.85	0.97	0

$z_{11} = 1$ *e* $1 \in I^c \Rightarrow$ *executar o algoritmo 4.4 para os pares (i, t)= (1, 2), (1, 3) e (1, 4).*

Para o par (i, t) = (1, 2), após a execução do algoritmo 4.4 obteve-se: $v_{12} = 0$, $\pi_{112} = 9$, $\pi_{122} = 40$, $\pi_{152} = 87.466$ *e* $\pi_{162} = 11.6$.

Para o par (i, t) = (1, 3), após a execução do algoritmo 4.4 obteve-se: $v_{13} = 0.252$ *e* $\pi_{113} = 6.116$.

Para o par (i, t) = (1, 4), como $v_{14} = 0$ *nada se pode diminuir.*

z_{23} = *1 e* $2 \in I^c \Rightarrow$ *executar o algoritmo 4.4 para o par (i, t)= (2, 4).*

Para o par (i, t) = (2, 4), após a execução do algoritmo 4.4 tem-se: $v_{24} = 0$, $\pi_{214} = 6$, $\pi_{224} = 39$, $\pi_{264} = 62$ *e* $\pi_{274} = 4$.

Como $\sum_{t=2}^{3} z_{3t}$ = *0 e 3* $\in I^o \Rightarrow$ *executar o algoritmo 4.4 para os pares (i, t)= (3, 2) e (3, 3).*

Para o par (i, t) = (3, 2), após a execução do algoritmo 4.4 obtemos: $v_{32} = 2.769$, $\pi_{352} = 206.918$ *e* $\pi_{382} = 342.837$.

Para o par (i, t) = (3, 3), após a execução do algoritmo 4.4 obtemos: $v_{33} = 0.112$, $\pi_{353} = 72.256$ *e* $\pi_{383} = 304.248$.

Como $\sum_{t=2}^{4} z_{5t}$ = *0 e 5* $\in I^o \Rightarrow$ *executar o algoritmo 4.4 para os pares (i, t)= (5, 2), (5, 3) e (5, 4).*

Para o par (i, t) = (5, 2), após a execução do algoritmo 4.4 obtém-se: $v_{52} = 0$, $\pi_{532} = 48.466$, $\pi_{552} = 265.466$ *e* $\pi_{582} = 470.466$.

Para o par (i, t) = (5, 3), após a execução do algoritmo 4.4 obtém-se: $v_{53} = 0$, $\pi_{553} = 20.312$ *e* $\pi_{583} = 100.896$.

Para o par (i, t) = (5, 4), como $v_{54} = 0$ *nada se pode diminuir.*

Resume-se em seguida a informação dual necessária para a construção do corte reforçado a introduzir em PM_R.

v_{it}	*1*	*2*	*3*	*4*
1	*0*	*0*	*0.252*	*0*
2	*2.516*	*4.066*	*0.624*	*0*
3	*4.92*	*2.769*	*0.112*	*0*
4	*4.111*	*0*	*0*	*1*
5	*3.739*	*0*	*0*	*0*

$$\sum_{j \in J} \sum_{t \in TP} \lambda_{jt} = 9\ 701.296$$

$\pi_{112} = 9$, $\pi_{113} = 6.116$, $\pi_{114} = 11$, $\pi_{122} = 40$, $\pi_{152} = 87.466$, $\pi_{162} = 11.6$, $\pi_{214} = 6$, $\pi_{224} = 39$, $\pi_{232} = 7.533$, $\pi_{252} = 209.533$, $\pi_{262} = 58.4$, $\pi_{264} = 62$, $\pi_{274} = 4$, $\pi_{282} = 100.533$, $\pi_{352} = 206.918$, $\pi_{353} = 72.256$, $\pi_{382} = 342.837$, $\pi_{383} = 304.248$, $\pi_{532} = 48.466$, $\pi_{552} = 265.466$, $\pi_{553} = 20.312$, $\pi_{582} = 470.466$, $\pi_{583} = 100.896$ *e e os restantes* π_{ijt} *iguais a 0.*

O corte a introduzir é o seguinte:

$$\rho \geq 9701.296 - (0 + 148.066)(1 - z_{11}) - (0.252 \times 249 + 6.116)(1 - z_{11} - z_{12}) - (0 + 11)(1 - z_{11} - z_{12} - z_{13}) - (2.516 \times 240 + 0) - (4.066 \times 240 + 375.996)(1 - z_{21}) - (0.624 \times 240 + 0)(1 - z_{21} - z_{22}) - (0 + 111)(1 - z_{21} - z_{22} - z_{23}) - (2.769 \times 192 + 549.755)\, z_{32} - (0.112 \times 192 + 376.504)(z_{32} + z_{33}) - (1 \times 435 + 0)(z_{42} - z_{43} - z_{44}) - (0 + 784.398)\, z_{52} - (0 + 121.208)(z_{52} + z_{53})$$

$\Leftrightarrow$

$$\rho \geq 9701.296 - 148.06(1 - z_{11}) - 68.864(1 - z_{11} - z_{12}) - 11(1 - z_{11} - z_{12} - z_{13}) - 603.84 - 1351.836(1 - z_{21}) - 149.76(1 - z_{21} - z_{22}) - 111(1 - z_{21} - z_{22} - z_{23}) - 1081.403\, z_{32} - 398.008(z_{32} + z_{33}) - 435(z_{42} - z_{43} - z_{44}) - 784.398\, z_{52} - 121.208(z_{52} + z_{53})$$

$\Leftrightarrow$

$$\rho \geq 7256.926 + 227.93\, z_{11} + 79.864\, z_{12} + 11\, z_{13} + 1612.6\, z_{21} + 260.76\, z_{22} - 111\, z_{23} - 1479.411\, z_{32} - 398.008\, z_{33} - 435\, z_{42} - 435\, z_{43} - 435\, z_{44} - 905.606\, z_{52} - 121.208\, z_{53}$$

nCortes = 2

Passo 1:

Após a introdução deste corte no problema mestre relaxado, este produz a seguinte configuração:

	1	2	3	4
1	■	□	□	□
2	■	■	■	□
3	□	□	□	■
4	□	■	■	■
5	□	□	□	□

$z_{11} = z_{23} = z_{34} = z_{42} = 1$
e para os restantes $z_{it} = 0$

$LI = V(PM_R) = 28\ 973.665$

Resolvendo DSP_z *com a configuração obtida, obtém-se:*

$V(DSP_z) = 28\ 973.665$

$LS = \min\{\ 28\ 973.665,\ 28\ 973.665\ \} = 28\ 973.665$

Passo 2:

Critério de paragem: $\frac{LS-LI}{LS} = \frac{28973.665-28973.665}{28973.665} = 0 < 0.001 \Rightarrow$ ***Passo 4***

Passo 4

Stop. A configuração dada por PM_R associada ao limite superior, LS, é óptima para o problema original. $\diamondsuit$

Bastaram dois cortes para obter a solução óptima para esta pequena instância. No entanto, neste exemplo, a configuração obtida com a resolução do primeiro PM_R é, na verdade, a configuração óptima. Numa instância de média ou grande dimensão, adivinham-se ganhos que podem ser bastante significativos, nomeadamente no que se refere ao número de cortes gerados.

4.4.2 Experiência computacional

Apresentam-se de seguida os tempos de *CPU* obtidos na sequência da aplicação do algoritmo da decomposição de Benders ao mesmo conjunto de instâncias ultilizadas nos testes computacionais já referidos neste capítulo e no anterior, usando o algoritmo heurístico 4.5 para reforço do corte usual de Benders. A tabela 4.13 apresenta os resultados obtidos para o grupo de instâncias com 10 localizações. As tabelas 4.14 e 4.15 apresentam os tempos obtidos para as instâncias com 20 e 50 localizações, respectivamente. Tal como antes, são apresentados os tempos de execução e o respectivo número de cortes necessário para a obtenção da solução óptima.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		3	3.48	1	0.16	1	0.09	1
5		5	0.38	3	0.27	3	0.25	3
		8	0.31	2	0.27	3	0.17	2
		3	0.82	4	0.63	4	0.52	3
10	20	5	0.89	4	0.73	4	0.81	4
		8	0.72	2	0.60	3	0.54	2
		3	11.25	5	24.66	6	17.14	6
15		5	889.94	9	831.58	7	760.37	8
		8	1.14	3	0.58	3	0.44	2
		3	0.75	1	0.22	2	0.11	1
5		5	0.55	3	0.36	3	0.22	2
		8	0.81	4	0.77	5	0.70	5
		3	1.83	4	1.14	3	0.72	2
10	50	5	1.25	2	0.56	3	0.44	2
		8	3.68	6	6.50	9	2.01	7
		3	26.41	7	26.72	8	22.56	7
15		5	5.56	3	11.50	4	14.00	4
		8	75.86	8	96.50	9	70.03	10
		3	0.75	2	0.44	3	0.29	2
5		5	0.62	1	0.50	1	0.26	2
		8	0.80	1	0.37	2	0.19	1
		3	3.00	4	1.51	4	1.08	3
10	100	5	3.09	3	1.55	3	1.17	2
		8	2.56	2	1.00	2	0.58	1
		3	10.30	4	3.64	4	4.80	4
15		5	19.78	4	24.95	5	18.31	5
		8	13.47	4	4.92	4	3.44	3

Tabela 4.13: Tempo de *CPU* (seg) e *#cortes* - 10 localizações.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	#*J*	#I^c	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		8	5.44	1	0.33	2	0.17	1
5		10	1.16	1	0.34	2	0.19	1
		16	4.11	12	2.27	1	2.84	13
		8	15.02	6	19.19	8	39.16	7
10	50	10	87.55	9	160.09	10	143.98	12
		16	89.30	7	73.17	8	39.08	8
		8	1371.70	10	4190.70	12	5264.95	11
15		10	1702.04	15	2115.70	19	3697.13	17
		16	1803.64	9	1553.50	11	2449.90	10
		8	9.61	4	1.61	5	1.46	4
5		10	5.33	2	1.12	3	0.77	2
		16	11.53	9	6.16	11	5.81	10
		8	28.92	7	11.53	6	11.34	6
10	100	10	40.59	9	17.27	9	15.36	8
		16	48.89	7	30.94	9	36.42	9
		8	210.10	8	117.52	7	75.75	6
15		10	8805.36	11	9417.92	9	36381.49	10
		16	689.08	6	433.24	7	341.89	5

Tabela 4.14: Tempo de *CPU* (seg) e #*cortes* - 20 localizações.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	#*J*	#I^c	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		20	87.00	2	2.97	2	3.09	2
5		25	101.16	3	6.13	4	6.53	4
		45	139.11	14	37.35	16	35.50	15
		20	990.21	15	113.48	6	4220.28	14
10	100	25	11502.53	11	1925.70	14	6170.55	12
		45	771.69	12	143.20	18	202.87	23
		20	781.75	2	34.01	3	349.72	3
15		25	157.17	2	9.74	1	1.97	1
		45	1682.20	10	246.80	10	996.59	9

Tabela 4.15: Tempo de *CPU* (seg) e #*cortes* - 50 localizações.

À semelhança do que foi feito com os resultados apresentados em secções anteriores, procede-se a uma síntese dos resultados a partir da qual se farão algumas considerações. Comecemos pelas instâncias com 10 localizações.

variante	tempo	cortes
Inicialização 1	40.00	3.6
Inicialização 2	38.62	4.0
Inicialização 3	34.12	3.5

Tabela 4.16: Valores médios - 10 localizações.

Os resultados da tabela 4.16 mostram que há realmente uma melhoria substancial ao nível dos tempos médios de CPU relativamente ao corte usual de Benders. Com a utilização desta estratégia também se alcançaram melhores tempos de CPU que os conseguidos utilizando a estratégia utilizada na secção anterior. Apenas a inicialização 1 apresenta um tempo médio superior ao obtido quando usados no fortalecimento, problemas de programação linear (tabela 4.10 - página 82), mas comparando as tabelas 4.7 (página 80) e 4.13, observa-se que tal se deve apenas a uma instância (20 clientes/15 períodos/$\#I = 5$) pois comparando os resultados, verifica-se um decréscimo nos tempos obtidos: considerando a inicialização 1, em cerca de 90 % das instâncias o tempo diminuiu acentuando-se esta diminuição nas instâncias que consideram 15 períodos de tempo. Nas inicializações 2 e 3, esta percentagem ultrapassa os 90 %, apesar de o número médio de cortes na segunda variante ter aumentado. Curiosamente, ao contrário do que vem sendo observado, a variante 3 apresenta tempos, em geral, inferiores à variante 2 que tem sido a que se tem revelado mais rápida. O número médio de cortes gerados não é sustancialmente diferente entre cada uma das variantes.

A tabela seguinte apresenta os valores médios dos resultados do conjunto de instâncias com 20 localizações.

variante	tempo	cortes
Inicialização 1	829.41	7.4
Inicialização 2	1008.48	7.7
Inicialização 3	2694.87	7.8

Tabela 4.17: Valores médios - 20 localizações.

No grupo com 20 localizações a variante de inicialização 1 apresenta tempo médio de *CPU* inferior às restantes variantes. No entanto, observando a tabela 4.14 verifica-se que em mais de 70 % das instâncias que constituem este grupo, a variante 2 apresenta tempos de *CPU* inferiores à inicialização 1. O número médio de cortes gerados por cada uma das variantes é semelhante, sendo a inicialização 1 a que os produziu em menor número. Comparativamente aos resultados obtidos com a utilização do problema PR_{it} no reforço do corte (comparando as tabelas 4.8 (página 81) e 4.14), podemos realçar, desde já, o facto de terem sido resolvidas todas as instâncias sem que nenhuma ultrapassasse as 12 horas. Existem diferenças bastante acentuadas nos tempos de *CPU*. A tabela 4.17 que apresenta os resultados médios, mostra isso claramente. O número de cortes gerados para cada instância é muito semelhante entre as duas estratégias de reforço. Vejamos se para o grupo com 50 localizações se continuam a verificar melhorias.

variante	tempo	cortes
Inicialização 1	1801.42	7.9
Inicialização 2	279.93	8.2
Inicialização 3	1331.90	9.2

Tabela 4.18: Valores médios - 50 localizações.

A variante de inicialização 2 apresenta tempo médio de *CPU* sustancialmente inferior às variantes de inicialização 1 e 3, mesmo tendo gerado, em média, maior número de cortes que a variante de inicialização 1. Comparativamente à estratégia da secção anterior e comparando as tabelas 4.12 (página 83) e 4.18 verifica-se mais vez um equilíbrio no número médio de cortes gerados. Mas relativamente ao tempo médio há uma melhoria significativa quando considerada a inicialização 2 em cada uma das estratégias. Todas as instâncias foram resolvidas em tempo inferior a 12 horas.

Tendo em conta os resultados obtidos, é evidente a vantagem de se construir, em cada iteração, um corte reforçado utilizando o algoritmo 4.5 (página 87) em vez do algoritmo 4.2 (página 74). A possível resolução de um elevado número de problemas PR_{it} em cada iteração pode conduzir a tempos superiores aos produzidos quando utilizado um algoritmo heurístico como o algoritmo 4.4.

4.5 Decomposição de Benders tendo por base $PLMC_3$

Muito embora os resultados obtidos pela aplicação da decomposição de Benders tendo como base a formulação $PLMC_4$, se tenham revelado muito encorajadores, nada impede que continuemos a tentar melhorá-los explorando outras estratégias para obter a solução óptima do problema de *phase-in/phase-out* em estudo nesta tese. Verificou-se na secção anterior que o facto de se ter que resolver um elevado número de problemas de programação linear no reforço do corte apresentado na secção 4.3 conduziu a tempos superiores aos tempos obtidos com a substituição por um procedimento heurístico para reforço do corte. No entanto não podemos esquecer o facto de que no algoritmo 4.6 é necessário resolver, em cada iteração, o dual de dois subproblemas: SP_z com e sem as restrições de afectação, sendo este último resolvido no algoritmo 4.5. Tal acontece porque o reforço é feito a partir do corte usual obtido com base em $PLMC_4$. Se o reforço for feito com base em $PLMC_3$, só será necessária a resolução, em cada iteração, do dual de SP_z (que naturalmente já não tem em conta as restrições de afectação).

Sugere-se nesta secção uma decomposição com base no modelo $PLMC_3$ utilizando para reforço do corte a mesma heurística utilizada na secção anterior.

4.5.1 Corte usual a partir da formulação $PLMC_3$

O corte usual de Benders gerado a partir da formulação $PLMC_3$ é o seguinte:

$$\rho \geq \sum_{j \in J} \sum_{t \in TP} \lambda_{jt} - \sum_{i \in I^c} \sum_{t \in TP} (v_{it}\, Q_i) \left(1 - \sum_{\tau=1}^{t-1} z_{i\tau}\right) - \sum_{i \in I^o} \sum_{t \in TP} (v_{it}\, Q_i) \left(\sum_{\tau=2}^{t} z_{i\tau}\right)$$

$$(\lambda,\, \pi,\, v) \in \Lambda \Pi V \qquad (4.28)$$

Esta desigualdade é derivada à semelhança de (4.21) (página 53) mas, agora, ignorando as variáveis π_{ijt}. Tal como aconteceu com o modelo $PLMC_4$, os resultados obtidos com a aplicação do corte usual acabam por satisfazer pouco. O número de cortes gerados é demasiado elevado conduzindo a tempos de CPU incomportáveis. Os resultados serão apresentados no Apêndice.

Tal como descrito na secção 4.3, podemos obter um corte mais forte resolvendo para os pares (i, t) nas condições apresentadas na subsecção 4.3.1 (página 71), o seguinte problema de programação linear:

(PR'_{it})

$$\min_{v_{it} \geq 0} \quad v_{it}\, Q_i \qquad (4.29)$$

$$s.a: \quad \lambda_{jt} - d_{jt}\, v_{it} \leq c_{ijt} \qquad j \in J \qquad (4.30)$$

Naturalmente que fortalecendo o corte usual gerado a partir do modelo $PLMC_3$, obtiveram-se melhorias relativamente aos resultados obtidos com o corte usual (estes resultados estão também disponíveis no Apêndice). No entanto, os resultados ficaram aquém dos melhores resultados obtidos na secção 4.4.

Uma vez que os conjuntos de restrições (4.3) e (4.4) são redundantes no problema inteiro, temos a livre opção de usar essas restrições. Podemos, inclusivamente, usar apenas algumas delas, de acordo com a nossa conveniência. Sugere-se de seguida um reforço do corte (4.28) usando algumas variáveis duais associadas aos conjuntos de restrições (4.3) e (4.4). Assim, a partir do corte usual, é aplicado o algoritmo 4.7 abaixo, que resulta do algoritmo 4.5 retirando a resolução de DSP_z. Depois de selecionado um par (i, t) é aplicado o algoritmo 4.4 (página 86) procurando-se, desta forma, obter novos valores para as variáveis v_{it} fazendo surgir algumas variáveis π_{ijt} com o intuito de reforçar o corte usual.

Uma desvantagem desta estratégia é o facto de o corte obtido poder ser mais fraco que o corte gerado através do reforço heurístico apresentado na secção 4.4, o que muito provavelmente acontece, principalmente com os cortes gerados nas primeiras iterações. À medida que o número de iterações aumenta, também aumenta no corte o número de variáveis π_{ijt} não nulas. Uma possível vantagem é que, sendo necessária apenas a resolução de um subproblema em cada iteração, tal pode conduzir a ganhos a nível dos tempos de *CPU*.

O algoritmo de reforço pode ser expresso de acordo com o seguinte procedimento:

Algorithm 4.7 Procedimento de reforço do corte gerado por $PLMC_3$

for $i \in I^c$ **do**
  Se existir τ tal que $z_{i\tau} = 1$
  for t $= \tau + 1, \ldots, T$ **do**
    if $v_{it} > 0$ **then**
      Executar o algoritmo 4.4
for $i \in I^o$ **do**
  if $\sum_{\tau=2}^{T} z_{i\tau} = 0$ **then**
    for $t = 2, \ldots, T$ **do**
      if $v_{it} > 0$ **then**
        Executar o algoritmo 4.4
  else
    Determinar τ tal que $z_{i\tau} = 1$
    for $t = 2, \ldots, \tau - 1$ **do**
      if $v_{it} > 0$ **then**
        Executar o algoritmo 4.4

O algoritmo da decomposição de Benders permanence tal como descrito no algoritmo 4.6.

A seguir apresenta-se um exemplo de aplicação usando a estratégia de reforço apresentada nesta subsecção, usando para tal, a mesma instância que foi utilizada nos exemplos anteriormente apresentados neste capítulo.

Exemplo 4.4 *Considere-se novamente a instância apresentada no exemplo (4.1) na página 57. Considere-se, mais uma vez, a forma de inicialização 1 e assuma-se, novamente,* $\varepsilon = 0.001$.

$n = 5$, $m = 10$, $I^c = \{1,2\}$, $I^o = \{3,4,5\}$ e $TP = \{1,2,3,4\}$.

A configuração admissível inicial é então:

	1	2	3	4
1	■	□	□	□
2	■	■	■	■
3	□	□	□	□
4	□	■	■	■
5	□	□	□	□

$z_{11} = z_{42} = 1$
e z_{it} = 0 para os retantes

Valor = 30 080.66

LI = -∞

LS = 30 080.66 (valor fornecido pela heurística)

Resolvendo DSP_z, obtém-se a seguinte solução dual:

λ_{jt}	1	2	3	4
1	4	13	10.872	9
2	108	149	46.72	26
3	208.096	258.466	132.464	11
4	787	455	472	25
5	701.54	787.466	513.312	56
6	86.87	126.6	186	32
7	113	62	66	64
8	864.516	1001.466	668.89	64
9	99	62	61	58
10	233	71	70	75

v_{it}	1	2	3	4
1	0	3	2.29	1.666
2	2.516	4.066	0.624	0
3	4.92	6.3	3.037	0
4	4.111	0	0	0
5	3.739	4.85	0.97	0

$$\sum_{j\in J}\sum_{t\in TP}\lambda_{jt} = 8\ 838.278$$

O corte usual gerado por esta solução é o seguinte:

$$\rho \geq 8838.278 - 747(1-z_{11}) - 570(1-z_{11}-z_{12}) - 414.8(1-z_{11}-z_{12}-z_{13}) -$$
$$603.84 - 975.84(1-z_{21}) - 149.76(1-z_{21}-z_{22}) - 1209.6\, z_{32} - 583.104(z_{32}+z_{33}) -$$
$$1217.35\, z_{52} - 243.47(z_{52}+z_{53})$$

Segue-se agora a fase de reforço aplicando o algoritmo 4.7, selecionando os pares (i, t) para os quais deve ser executado o algoritmo 4.4. Contudo, repare-se que, tendo o processo sido inicializado com a solução da heurística, os valores das variáveis z_{it}, $i \in I$ e $t \in TP$, são iguais aos do exemplo 4.3 (página 89). Como tal, os pares (i, t) selecionados estão de acordo com a selecção feita no referido exemplo. Após a aplicação do algoritmo 4.4 (de alteração das variáveis v_{it} e alguns π_{ijt}, $j \in J$), obtêm-se os seguintes valores:

v_{it}	1	2	3	4
1	0	0	0.252	0
2	2.516	4.066	0.624	0
3	4.92	2.769	0.112	0
4	4.111	0	0	0
5	3.739	0	0	0

$\pi_{112} = 9$, $\pi_{113} = 6.116$, $\pi_{114} = 5$, $\pi_{122} = 40$, $\pi_{152} = 87.466$, $\pi_{162} = 11.6$,
$\pi_{352} = 206.918$, $\pi_{353} = 72.256$, $\pi_{382} = 342.837$, $\pi_{383} = 304.248$, $\pi_{532} = 48.466$,
$\pi_{552} = 265.466$, $\pi_{553} = 20.312$, $\pi_{582} = 470.466$, $\pi_{583} = 100.896$, *sendo os restantes* π_{ijt} *iguais a 0.*

A diferença relativamente ao referido exemplo é que existem neste caso menos variáveis π_{ijt} não nulas, que são apenas as que resultaram da aplicação do algoritmo de reforço. O corte gerado por esta informação traduz-se na seguinte desigualdade:

$$\rho \geq 8838.278 - 148.066(1 - z_{11}) - 68.864(1 - z_{11} - z_{12}) - 5(1 - z_{11} - z_{12} - z_{13}) -$$
$$603.84 - 975.84(1 - z_{21}) - 149.76(1 - z_{21} - z_{22}) - 1081.403\, z_{32} - 398.008(z_{32} + z_{33}) -$$
$$784.398\, z_{52} - 121.208(z_{52} + z_{53})$$

Repare-se que se $z_{21} = 1$, o membro direito desta desigualdade é igual ao membro direito do primeiro corte reforçado do exemplo 4.3.

nCortes = 1

Passo 1:

A resolução de PM_R conduz à seguinte configuração:

	1	2	3	4
1	■	■	■	□
2	■	□	□	□
3	□	□	□	■
4	□	■	■	■
5	□	□	□	□

$z_{13} = z_{21} = z_{34} = z_{42} = 1$
e para os restantes $z_{it} = 0$

$LI = V(PM_R) = 28\ 669.494$

*Fixando o vector **z** obtido a partir da resolução do problema mestre relaxado, resolve-se DSP_z conduzindo à seguinte solução dual:*

$V(DSP_z)$ = 29 551.81

LS = min { 30 080.66, 29 551.81 } = 29 551.81

$\sum_{j\in J}\sum_{t\in TP}\lambda_{jt}$ = *10 001.432*

Passo 2:

Critério de paragem: $\frac{LS-LI}{LS} = \frac{29551.81-28669.494}{29551.81} = 0.02985 > 0.001$

Passo 3:

Após a aplicação do algoritmo 4.5, obtém-se:

v_{it}	1	2	3	4
1	0	0	0.252	0
2	2.516	2.733	0.624	0
3	4.92	2.972	3.944	0
4	4.111	0	0	1
5	3.739	0	0	0

$\pi_{112} = 9$, $\pi_{113} = 6.116$, $\pi_{114} = 1$, $\pi_{122} = 40$, $\pi_{152} = 87.466$, $\pi_{162} = 11.6$, $\pi_{214} = 6$, $\pi_{223} = 78.84$, $\pi_{224} = 39$, $\pi_{232} = 56.866$, $\pi_{233} = 151.536$, $\pi_{252} = 1.866$, $\pi_{253} = 257.564$, $\pi_{262} = 20.4$, $\pi_{264} = 62$, $\pi_{274} = 4$, $\pi_{282} = 162.866$, $\pi_{283} = 301.312$, $\pi_{352} = 105.81$, $\pi_{353} = 88.376$, $\pi_{382} = 356.62$, $\pi_{383} = 206.98$, $\pi_{532} = 56$, $\pi_{533} = 83$, $\pi_{552} = 178$, $\pi_{553} = 277.876$, $\pi_{582} = 504$, $\pi_{583} = 402.208$ *e para os restantes* π_{ijt} *iguais a 0.*

Esta informação conduz ao próximo corte de Benders a introduzir em PM_R.

$$\rho \geq 10001.432 - (0+148.066)(1-z_{11}) - (0.252\times 249+6.116)(1-z_{11}-z_{12}) - 1(1-z_{11}-z_{12}-z_{13}) - (2.516\times 240+0) - (2.733\times 240+241.998)(1-z_{21}) - (0.624\times 240+789.252)(1-z_{21}-z_{22}) - (0+111)(1-z_{21}-z_{22}-z_{23}) - (2.972\times 192+462.43)\,z_{32} - (3.944\times 192+295.356)(z_{32}+z_{33}) - 1\times 435(z_{42}+z_{43}+z_{44}) - (0+738)\,z_{52} - (0+763.084)(z_{52}+z_{53})$$

$\Leftrightarrow$

$$\rho \geq 10001.432 - 148.066(1-z_{11}) - 68.864(1-z_{11}-z_{12}) - 1(1-z_{11}-z_{12}-z_{13}) - 603.84 - 897.918(1-z_{21}) - 939.012(1-z_{21}-z_{22}) - 111(1-z_{21}-z_{22}-z_{23}) - 1033.054\,z_{32} - 1052.68(z_{32}+z_{33}) - 435(z_{42}+z_{43}+z_{44}) - 738\,z_{52} - 763.084(z_{52}+z_{53})$$

$\Leftrightarrow$

$$\rho \geq 7231.732 + 217.93\,z_{11} + 69.864\,z_{12} + \ z_{13} + 1947.93\,z_{21} + 1050.012\,z_{22} + 111\,z_{23} - 2085.658\,z_{32} - 1052.604\,z_{33} - 435\,z_{42} - 435\,z_{43} - 435\,z_{44} - 1501.084\,z_{52} - 763.084\,z_{53}$$

nCortes = *2*

Passo 1:

Ao adicionar o corte reforçado a PM_R*, a resolução deste último produz a seguinte configuração:*

	1	2	3	4
1	■	□	□	□
2	■	■	■	□
3	□	□	□	■
4	□	■	■	■
5	□	□	□	□

$z_{11} = z_{23} = z_{34} = z_{42} = 1$
e para os restantes $z_{it} = 0$

$LI = V(PM_R) = 28\,973.665$

Pela resolução de DSP_z *obtém-se,*

$V(DSP_z) = 28\,973.665$

$LS = min\ \{\ 29\,551.81,\ 28\,973.665\ \} = 28973.665$

Passo 2:

Critério de paragem: $\frac{LS-LI}{LS} = 0 < 0.001 \Rightarrow$ ***Passo 4***

Passo 4

Stop. A configuração dada por PM_R *associada ao limite superior, LS, é óptima para o problema original.* ♢

Observação *Note-se que, apesar de o reforço ter sido feito a partir do corte usual da formulação* $PLMC_3$*, foram suficientes 2 cortes para obter a solução óptima para esta instância.* ♣

4.5.2 Experiência computacional

A seguir apresentam-se os resultados obtidos, quando aplicada a estratégia de reforço de um corte apresentada na subsecção anterior, sendo testadas as três variantes de inicialização: 1, 2 e 3. Na tabela 4.19 apresentam-se os resultados das instâncias com 10 localizações. Da mesma forma que em secções anteriores, são descriminados os tempos de execução do algoritmo utilizando cada uma das variantes de inicialização e o número de cortes gerados até à convergência. As tabelas 4.20 e 4.21 apresentam a mesma informação mas para as instâncias em que são consideradas 20 e 50 localizações, respectivamente.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		3	0.08	1	0.03	1	0.09	1
5		5	0.20	3	0.17	3	0.25	3
		8	0.23	3	0.14	3	0.21	2
		3	0.45	3	0.34	3	0.53	3
10	20	5	0.84	3	0.70	3	0.78	4
		8	0.63	2	0.42	2	0.53	2
		3	1.72	3	1.17	3	16.89	6
15		5	27.13	4	2.17	3	759.20	8
		8	3.80	2	0.45	2	0.48	2
		3	0.91	1	0.08	1	0.16	1
5		5	1.25	3	0.26	1	0.27	2
		8	1.55	3	0.34	3	0.81	5
		3	4.49	2	0.70	2	0.86	2
10	50	5	3.87	2	1.95	2	0.52	2
		8	4.19	3	0.47	3	2.34	7
		3	8.02	3	7.35	3	18.51	7
15		5	14.75	3	2.55	3	15.09	4
		8	30.05	3	18.45	3	71.20	10
		3	2.69	2	0.25	2	0.49	2
5		5	2.88	1	0.14	1	0.47	2
		8	3.02	1	0.12	1	0.28	1
		3	10.67	3	0.64	3	1.42	3
10	100	5	8.60	2	0.60	2	1.39	2
		8	9.28	1	0.24	1	0.70	1
		3	27.47	3	1.61	3	5.49	4
15		5	23.20	3	1.98	3	19.09	5
		8	36.81	3	1.25	3	4.27	3

Tabela 4.19: Tempo de *CPU* (seg) e *#cortes* - 10 localizações.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		8	9.62	1	0.12	1	0.22	1
5		10	7.30	1	0.11	1	0.22	1
		16	11.05	3	0.42	3	3.49	13
		8	37.94	3	1.22	3	39.48	7
10	50	10	49.11	2	0.53	2	147.26	12
		16	62.53	6	7.01	6	39.39	8
		8	155.58	3	245.20	3	5232.88	11
15		10	525.78	16	96.14	8	3671.52	17
		16	102.69	3	64.34	4	2455.29	10
		8	31.55	4	0.82	4	1.72	4
5		10	26.69	2	0.41	2	1.00	2
		16	22.83	3	0.61	3	6.48	10
		8	110.34	3	1.59	3	12.08	6
10	100	10	140.91	3	1.47	3	16.89	8
		16	110.07	5	4.65	5	41.72	9
		8	328.33	4	9.62	4	78.17	6
15		10	383.80	4	2487.36	7	35375.88	10
		16	213.09	2	9.61	2	344.34	5

Tabela 4.20: Tempo de *CPU* (seg) e *#cortes* - 20 localizações.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		20	84.44	2	0.64	2	2.95	2
5		25	94.06	4	1.11	4	6.56	4
		45	110.77	11	4.04	11	34.93	15
		20	782.49	38	325.69	28	4525.01	14
10	100	25	1084.93	25	241.58	16	10038.79	14
		45	667.27	10	22.75	20	200.06	23
		20	672.17	2	2.64	2	339.07	3
15		25	*out mem.*	-	5.53	1	*out mem.*	-
		45	1324.48	2	305.51	3	987.39	9

Tabela 4.21: Tempo de *CPU* (seg) e *#cortes* - 50 localizações.

⌜ *Para a instância assinalada com 'out mem.' na tabela 4.21, o solver apresentou um erro de memória aquando da resolução de PM_R.* ⌟

Com base na tabela 4.19, apresentam-se os valores médios para as instâncias com 10 localizações:

variante	tempo	cortes
Inicialização 1	8.47	2.4
Inicialização 2	1.65	2.3
Inicialização 3	34.16	3.5

Tabela 4.22: Valores médios - 10 localizações.

Através da tabela 4.22, podemos constatar que a segunda variante de inicialização faz com que o algoritmo de Benders tenha uma melhor *performance* do que considerando cada uma das outras variantes. Veja-se, através da tabela 4.19 que na totalidade das instâncias, a segunda variante de inicialização apresenta tempos de CPU inferiores às restantes exceptuando uma instância em que a inicialização 3 apresenta um tempo ligeiramente inferior à inicialização 2. O número médio de cortes gerados por esta última é também menor, apesar de, como já foi mencionado, o corte gerado nesta inicialização não tem possibilidade de reforço, sendo um mero corte usual.

A tabela 4.23 apresenta os valores médios para as instâncias com 20 localizações.

variante	tempo	cortes
Inicialização 1	129.40	3.8
Inicialização 2	162.84	3.6
Inicialização 3	2637.11	7.8

Tabela 4.23: Valores médios - 20 localizações.

Realça-se, neste conjunto de instâncias com 20 localizações, o número médio de cortes gerados com a utilização da inicialização 3, assim como o tempo médio produzido, para a qual a instância com 100 clientes/15 períodos/ $\#I^c$=10 foi especialmente difícil. Embora o tempo médio de CPU obtido usando a inicialização 2 seja superior ao tempo médio obtido usando a inicialização 1, na realidade, observando a tabela 4.20, verifica-se que esta última conduz quase sempre a tempos superiores à variante 2. Apesar de o tempo de CPU ser superior em algumas instâncias, a variante 2 gera um número de cortes inferior às restantes variantes de inicialização.

Na tabela seguinte apresentam-se os valores médios referentes às instâncias com 50 localizações. Tal como sucedeu em secções anteriores, não será tida em conta a instância para a qual o *general solver* apresentou um erro de memória.

variante	tempo	cortes
Inicialização 1	602.58	11.8
Inicialização 2	113.00	10.8
Inicialização 3	2016.85	10.5

Tabela 4.24: Valores médios - 50 localizações.

No conjunto das instâncias com 50 localizações, mais uma vez, se evidencia a boa *performance* da variante de inicialização 2, que conduz a tempos sempre inferiores (na maioria das instâncias, significativamente) às restantes. O número de cortes produzidos usando cada uma das variantes de inicialização, à semelhança do que tem acontecido também nas secções anteriores com a aplicação de outras estratégias de corte ao grupo de instâncias com 50 localizações, é especialmente superior quando se considera um horizonte de planeamento dividido em 10 períodos de tempo e não para 15 períodos como sucede com os restantes grupos de instâncias.

Relativamente aos tempos obtidos nas secções anteriores, não restam dúvidas de que até agora, é a melhor estratégia para obtenção de um corte. Observaram-se de facto, em todos os grupos de instâncias, reduções significativas nos tempos de *CPU* comparativamente à aplicação de outros cortes nas secções anteriores.

No que se refere ao número de cortes, esta estratégia gera, em termos médios, menor número de cortes nos grupos com 10 e 20 localizações. No entanto, a utilização dos cortes apresentados nas secções 4.3 e 4.4 geram, em média, menos cortes na resolução do grupo de instâncias com 50 localizações.

4.5.3 Uma outra estratégia para obtenção de um corte

Ainda considerando o corte baseado em $PLMC_3$, tentou-se uma pequena variante.

Na estratégia apresentada na subsecção 4.5.1, depois de efectuado o reforço, algumas variáveis π_{ijt} passam a tomar valor positivo. No corte da iteração seguinte estas mesmas variáveis mantêm o seu valor com excepção daquelas cujo par (i, t) é selecionado para reforço sendo o próprio procedimento (algoritmo 4.4) a reiniciá-las com valor 0, de forma a poder controlar-se o seu crescimento.

Nesta estratégia, que será designada por 2ª estratégia, o que se propõe é que, sempre que uma nova iteração se inicia, todas as variáveis π_{ijt} voltam a tomar valor nulo. Assim, nas restrições de corte consideramos, em cada iteração, apenas as variáveis $\pi_{ijt} > 0$ obtidas a partir da alteração, nessa mesma iteração, da variável v_{it} correspondente, considerando nulas todas as outras.

A alteração ao algoritmo de Benders com corte reforçado (algoritmo 4.6) é mínima. Basta que no início do **Passo 3**, antes da resolução de DSP_z se insira a condição $\pi_{ijt} = 0, \forall\ i \in I,\ j \in J$ e $t \in TP$, o que faz com que na inicialização do procedimento de alteração de uma variável v_{it}, algoritmo 4.4, se possa retirar a condição $\pi_{ijt} = 0, \forall\ j \in J$.

Mesmo não se tratando de um corte melhor que o induzido pela estratégia anterior, não é de desprezar a *performance* obtida por este, como se pode verificar no gráfico que se apresenta de seguida e que será confirmado adiante aquando da apresentação dos resultados obtidos.

O gráfico da figura 4.1 mostra para a instância gerada com 10 localizações, 50 clientes e 15 períodos de tempo, considerando 8 serviços existentes no início do horizonte de planeamento (variante de inicialização usando a solução da heurística), a sucessão de limites inferiores e superiores obtidos, quando aplicado o corte usual (4.28) e a estratégia de reforço acabada de descrever. No primeiro caso foram gerados 28 cortes resultando num consumo de *CPU* de mais de 1200 segundos. Usando a estratégia acima, foram gerados apenas 9 cortes em aproximadamente 135 segundos. De referir apenas que quando se utilizou o reforço proposto na secção anterior (algoritmo 4.7) foram gerados apenas 3 cortes tendo-se gasto cerca de 30 segundos no procedimento de decomposição.

Figura 4.1: Instância de 10 localizações, 50 clientes, 15 períodos e $\#I^c = 8$.

4.5.4 Experiência computacional

Apresentam-se, seguidamente, os resultados obtidos mediante a aplicação do algoritmo usando a estratégia de reforço proposta na subsecção anterior considerando as três variantes de inicialização.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		3	0.06	1	0.05	1	0.03	1
5		5	0.24	3	0.17	3	0.20	3
		8	0.16	3	0.09	3	0.12	3
		3	0.48	3	0.34	3	0.41	3
10	20	5	1.11	4	0.97	4	0.64	4
		8	0.59	2	0.37	2	0.44	2
		3	2.87	3	13.27	6	16.44	6
15		5	∗768.05	8	692.17	9	762.89	8
		8	3.67	2	0.33	2	0.38	2
		3	0.91	1	0.06	1	0.08	1
5		5	1.20	3	0.19	2	0.19	2
		8	1.77	5	0.55	5	0.53	5
		3	4.25	2	0.55	2	0.69	2
10	50	5	3.92	2	0.27	2	0.33	2
		8	7.19	5	3.34	5	1.64	7
		3	31.53	6	29.25	7	17.32	7
15		5	19.75	3	7.61	3	14.35	4
		8	∗135.82	9	94.92	9	69.75	10
		3	2.67	2	0.25	2	0.22	2
5		5	2.86	1	0.14	1	0.22	2
		8	3.00	1	0.13	1	0.11	1
		3	10.81	3	0.81	3	0.84	3
10	100	5	8.53	2	0.50	2	0.58	2
		8	9.30	1	0.24	1	0.28	1
		3	27.31	3	1.47	3	2.87	4
15		5	28.22	5	7.04	5	15.22	5
		8	38.42	3	2.86	3	1.83	3

Tabela 4.25: Tempo de *CPU* (seg) e *#cortes* - 10 localizações.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		8	9.63	1	0.13	1	0.12	1
5		10	7.31	1	0.11	1	0.12	1
		16	12.84	14	1.81	12	2.11	13
		8	51.66	6	38.29	6	38.18	7
10	50	10	179.01	9	139.70	8	143.55	12
		16	120.56	8	77.25	7	35.64	8
		8	1381.17	10	4475.78	10	5241.97	11
15		10	2199.45	18	5526.89	17	3675.72	17
		16	648.69	11	479.53	11	2425.58	10
		8	31.59	4	0.88	4	0.88	4
5		10	26.74	2	0.39	2	0.42	2
		16	24.50	10	1.97	8	2.25	10
		8	121.78	8	11.31	7	5.69	6
10	100	10	146.31	8	6.81	8	6.14	8
		16	125.91	9	15.39	7	32.22	9
		8	441.99	6	115.73	6	66.42	6
15		10	8473.19	12	10193.88	10	35639.24	10
		16	658.66	5	458.28	5	333.78	5

Tabela 4.26: Tempo de *CPU* (seg) e *#cortes* - 20 localizações.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		20	84.37	2	1.00	3	0.61	2
5		25	93.45	2	0.56	2	1.25	4
		45	113.83	18	6.92	15	7.08	15
		20	1477.48	13	453.16	15	4591.54	16
10	100	25	2082.24	15	998.00	17	9964.28	14
		45	690.06	12	23.80	11	67.11	23
		20	676.74	2	62.51	4	326.53	3
15		25	*out mem.*	-	4.28	1	*out mem.*	-
		45	1460.76	8	153.75	8	881.03	9

Tabela 4.27: Tempo de *CPU* (seg) e *#cortes* - 50 localizações.

⌜ *Para a instância assinalada com 'out mem.' na tabela 4.27, o solver apresentou um erro de memória aquando da resolução de* PM_R. ⌟

A tabela seguinte apresenta os valores médios do conjunto de instâncias com 10 localizações quando usadas cada uma das variantes de inicialização.

variante	tempo	cortes
Inicialização 1	41.28	3.2
Inicialização 2	31.78	3.3
Inicialização 3	33.65	3.5

Tabela 4.28: Valores médios - 10 localizações.

Relativamente à variante 3 de inicialização, isto é, iniciando com a resolução de PM_R sem qualquer corte, a *performance* mantém-se em relação à estratégia de reforço cujos resultados foram apresentados na subsecção 4.5.2. Contudo, verifica-se uma diminuição no tempo utilizado apesar da diferença ser pouco significativa pois, com excepção de uma instância, em todas as outras instâncias deste grupo verificou-se um pequeno decréscimo nos tempos relativamente à estratégia anterior quando considerada a variante 3. Embora o número médio de cortes gerados seja igual ao número médio de cortes gerados com a primeira estratégia, esta segunda estratégia conduziu a um tempo médio inferior à estratégia anterior apresentada na subsecção 4.5.2. Não será, portanto, de desprezar a *performance* obtida por esta última, nomeadamente nas intâncias de menor dimensão.

Em relação à primeira e segunda variantes de inicialização, os resultados alteraram-se um pouco. Foram gerados, em média, maior número de cortes cortes e o tempo médio de CPU aumentou significativamente. No entanto, pode verificar-se que esta diferença acentuada é causada essencialmente pelas duas instâncias assinaladas com '$*$' na tabela 4.25. Verifica-se ainda, comparando as tabelas 4.19 (página 108) e 4.25 que, em muitas das instâncias, esta estratégia consegue obter convergência com igual número de cortes da estratégia anterior. Frequentemente, quando tal acontece, o tempo de CPU é inferior.

Para o conjunto de instâncias de 20 localizações, os valores médios para cada uma das variantes de inicialização são os seguintes:

variante	tempo	cortes
Inicialização 1	814.50	7.9
Inicialização 2	1196.90	7.2
Inicialização 3	2647.22	7.8

Tabela 4.29: Valores médios - 20 localizações.

Neste grupo de instâncias, em que as dimensões começam a ser consideráveis, registou-se um aumento significativo nos tempos produzidos pela estratégia proposta na subsecção 4.5.3. A variante de inicialização 3 gerou, em média, igual número de cortes e o tempo médio utilizado também não se diferenciou muito em relação à estratégia cujos resultados foram apresentados na subsecção 4.5.2. Mas com as restantes variantes de inicialização registaram-se aumentos muito acentuados, quer no tempo médio de *CPU* gasto quer no número médio de cortes produzido. No entanto, pode dizer-se que o tempo de resolução de PM_R foi a causa destas grandes diferenças. Veja-se que as instâncias em que se considera T=15, em particular com $\#I^c$=8 e $\#I^c$=10, são as que oferecem maior dificuldade.

A tabela 4.30 resume os valores médios de cada uma das variantes de inicialização para o conjunto de instâncias com 50 localizações (não é considerada a instância para a qual o *general solver* apresentou um erro de memória aquando da resolução de PM_R).

variante	tempo	cortes
Inicialização 1	834.87	9.0
Inicialização 2	212.46	9.4
Inicialização 3	1979.93	10.8

Tabela 4.30: Valores médios - 50 localizações.

Também para a estratégia proposta na subsecção 4.5.3, a variante de inicialização 2 conduziu a um tempo médio de *CPU* significativamente inferior a qualquer das restantes variantes, sendo muito acentuada a diferença especialmente para as instâncias com 10 e 15 períodos de tempo.

Comparativamente aos resultados da subsecção 4.5.2, para este conjunto de instâncias, apenas a variante de inicialização 3 conseguiu um tempo médio inferior. Considerando as variantes de inicialização 1 e 2, foram gerados, em média, menor número de cortes e mesmo o tempo médio não é significativamente superior à primeira estratégia desta secção.

4.6 Conclusões

De acordo com todos os resultados obtidos ao longo deste capítulo mediante a aplicação dos diferentes tipos de corte e para cada uma das variantes de inicialização, verifica-se uma clara vantagem em considerar como solução inicial a que se obtém considerando todos os serviços abertos tanto quanto possível. Far-se-á nesta secção uma comparação em termos de estratégias de corte, tendo por base a variante de inicialização 2. Apresenta-se ainda na última coluna das tabelas, o melhor tempo produzido pelo procedimento de *Branch-and-Bound* do *IlogCplex* [13] por algum dos dos quatro modelos analisados no capítulo 3. Como vimos na altura, o melhor tempo foi obtido a partir do modelo $PLMC_3$ ou do modelo $PLMC_4$ (subsecção 3.2.5 - página 20).

			fortalecido (sec.4.3)		ref. Heur. (sec.4.4)		1ª Estrat. (sec.4.5)		2ª Estrat. (sec.4.5)		
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*	**Cplex**
		3	0.56	1	0.16	1	0.03	1	0.05	1	0.03
5		5	1.30	3	0.27	3	0.17	3	0.17	3	0.13
		8	1.05	3	0.27	3	0.14	3	0.09	3	0.09
		3	3.72	3	0.63	4	0.34	3	0.34	3	0.91
10	20	5	3.70	4	0.73	4	0.70	3	0.97	4	0.69
		8	1.77	2	0.60	3	0.42	2	0.37	2	0.69
		3	9.81	4	24.66	6	1.17	3	13.27	6	7.24
15		5	1130.83	7	831.58	7	2.17	3	692.17	9	543.67
		8	3.02	2	0.58	3	0.45	2	0.33	2	0.86
		3	0.66	1	0.22	2	0.08	1	0.06	1	0.06
5		5	1.53	3	0.36	3	0.26	1	0.19	2	0.30
		8	1.81	4	0.77	5	0.34	3	0.55	5	0.64
		3	2.84	2	1.14	3	0.70	2	0.55	2	6.11
10	50	5	2.10	2	0.56	3	1.95	2	0.27	2	0.67
		8	8.67	6	6.50	9	0.47	3	3.34	5	6.44
		3	53.85	7	26.72	8	7.35	3	29.25	7	67.11
15		5	15.39	3	11.50	4	2.55	3	7.61	3	86.95
		8	124.36	7	96.50	9	18.45	3	94.92	9	280.11
		3	1.81	2	0.44	3	0.25	2	0.25	2	0.39
5		5	0.75	1	0.50	1	0.14	1	0.14	1	0.08
		8	0.70	1	0.37	2	0.12	1	0.13	1	0.30
		3	4.72	3	1.51	4	0.64	3	0.81	3	13.83
10	100	5	3.31	2	1.55	3	0.60	2	0.50	2	4.83
		8	1.95	1	1.00	2	0.24	1	0.24	1	1.16
		3	9.20	3	3.64	4	1.61	3	1.47	3	104.05
15		5	24.13	4	24.95	5	1.98	3	7.04	5	240.69
		8	9.55	3	4.92	4	1.25	3	2.86	3	133.97

Tabela 4.31: Tempo de *CPU* (seg) e *#cortes* - 10 localizações.

Na tabela 4.32 apresenta-se o tempo médio consumido pelas instâncias com 10 localizações, usando cada uma das estratégias de reforço de um corte consideradas assim como o número médio de cortes gerados. Para efeitos de comparação, apresenta-se, ainda, o tempo médio consumido pela decomposição de Benders quando usado o corte usual com base na formulação $PLMC_4$ (apresentado na secção 4.3) e pelo procedimento de *Branch-and-Bound* do *IlogCplex 9.0* [13].

	tempo	cortes
corte usual	677.92	9.2
corte fortalecido	52.70	3.1
reforço Heur.	38.62	4.0
1ª Estratégia	1.65	2.3
2ª Estratégia	31.78	3.3
Cplex	55.62	-

Tabela 4.32: Valores médios - 10 localizações.

Podemos desde já constatar que a 1ª estratégia (subsecção 4.5.2) de reforço domina qualquer das outras relativamente ao tempo médio de *CPU* e ao número médio de cortes gerados, tendo resolvido este grupo de instâncias em cerca de 0.24 % do tempo médio consumido pelo corte usual, 4.3 % do corte reforçado heuristicamente, 3.1 % do corte fortalecido, 5.2 % da 2ª estratégia e em apenas 2.96 % do tempo total do procedimento *Branch-and-Bound* do *Ilog Cplex*. No entanto, para esta diferença tão significativa contribuiu muito a instância com 20 clientes/15 períodos/$\#I^c$=5, que se tornou particularmente difícil. Deste grupo, as instâncias com $\#I^c$=5 e $\#I^c$=8 consomem mais tempo do que as instâncias com $\#I^c$=3. Um maior número de períodos conduz a um maior número de variáveis inteiras, o que justifica que as instâncias que consideram 15 períodos sejam as que consomem mais tempo.

As tabelas 4.31 e 4.32 em comparação com as tabelas 4.1 (66) e $A1$ do Apêndice permitem-nos constatar que a aplicação da decomposição de Benders sem um reforço não oferece qualquer vantagem. O número de cortes produzido é, de facto, muito grande, o que faz com o procedimento *Branch-and-Bound* do *IlogCplex* [13] tenha um melhor desempenho. Tendo em conta os resultados, é evidente a necessidade de aplicar a decomposição de Benders usando cortes fortalecidos de alguma forma. Apesar de a 2ª estratégia não ser em geral melhor do que a 1ª estratégia, repare-se que se não se considerasse a instância com 20 clientes/15 períodos/$\#I^c = 5$, em que a 2ª estratégia precisou de 692.17 segundos para obter a solução óptima (de resto foi uma instância difícil até para o procedimento *Branch-and-Bound* do *Ilog Cplex*), o tempo total para a 2ª estratégia para a resolução deste grupo de instâncias, seria de apenas 165.77 segundos. Repare-se também que, tirando um ou outro caso, o número de cortes não é substancialmente superior ao número produzido pela 1ª estratégia e são várias as instâncias em que, produzindo o mesmo número de cortes que a 1ª estratégia resulta num tempo igual ou inferior.

			fortalecido (sec.4.3)		ref. Heur. (sec.4.4)		1ª Estrat. (sec.4.5)		2ª Estrat. (sec.4.6)		
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*	**Cplex**
		8	1.12	1	0.33	2	0.12	1	0.13	1	0.14
5		10	1.11	1	0.34	2	0.11	1	0.11	1	0.13
		16	13.13	13	2.27	1	0.42	3	1.81	12	1.02
		8	49.52	6	19.19	8	1.22	3	38.29	6	46.25
10	50	10	151.03	10	160.09	10	0.53	2	139.70	8	153.55
		16	73.97	7	73.17	8	7.01	6	77.25	7	46.72
		8	5586.60	10	4190.70	12	245.20	3	4475.78	10	1202.34
15		10	2072.72	20	2115.70	19	96.14	8	5526.89	17	848.45
		16	1612.20	10	1553.50	11	64.34	4	479.53	11	2808.12
		8	5.20	4	1.61	5	0.82	4	0.88	4	0.89
5		10	2.61	2	1.12	3	0.41	2	0.39	2	0.28
		16	10.95	8	6.16	11	0.61	3	1.97	8	2.12
		8	20.42	5	11.53	6	1.59	3	11.31	7	75.53
10	100	10	30.22	8	17.27	9	1.47	3	6.81	8	33.69
		16	41.95	7	30.94	9	4.65	5	15.39	7	150.67
		8	154.99	5	117.52	7	9.62	4	115.73	6	382.81
15		10	> 12h	-	9 417.92	9	2487.36	7	10193.88	10	> 12h
		16	477.49	6	433.24	7	9.61	2	458.28	5	4646.67

Tabela 4.33: Tempo de *CPU* (seg) e *#cortes* - 20 localizações.

	tempo	cortes
corte usual	2636.11	22.6
corte fortalecido	606.19	7.2
Reforço Heur.	1008.48	7.7
1ª Estratégia	162.85	3.6
2ª Estratégia	1196.89	7.2
Cplex	611.73	-

Tabela 4.34: Valores médios - 20 localizações.

Continua a verificar-se neste conjunto de instâncias o bom desempenho da decomposição de Benders quando utilizado o corte desenvolvido com base em $PLMC_3$ e com um reforço conseguido heuristicamente. A diferença entre os tempos de CPU é muito acentuada e o mesmo se verifica relativamente ao número de cortes gerados para obter a solução óptima. A tabela 4.34 mostra que esta estratégia permite resolver a totalidade das instâncias deste grupo em cerca de 16 % do tempo médio conseguido pelo corte obtido com base na formulação $PLMC_4$ sendo reforçado também heuristicamente e em apenas cerca de 13.6 % do tempo da 2ª estratégia. O *general solver* começa já neste conjunto de instâncias a revelar dificuldades, havendo, inclusivamente, uma instância para a qual não se obteve a solução óptima em tempo inferior a 12 horas (43 000 segundos).

Relativamente ao corte usual, ainda de acordo com a tabela 4.34, produz um número médio de cortes equivalente a 6 vezes mais do que denominada 1ª estratégia.

A seguir analisa-se o conjunto de instâncias com 50 localizações tendo em conta informação análoga à que entretanto foi apresentada para os conjuntos de instâncias com 10 e 20 localizações.

			fortalecido (sec.4.3)		ref. Heur. (sec.4.4)		1ª Estrat. (sec.4.5)		2ª Estrat. (sec.4.5)		
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*	**Cplex**
		20	8.56	2	2.97	2	0.64	2	1.00	3	0.83
5		25	12.48	3	6.13	4	1.11	4	0.56	2	2.11
		45	59.56	15	37.35	16	4.04	11	6.92	15	31.00
		20	3550.53	16	113.48	6	325.69	28	453.16	15	16516.25
10	100	25	2130.86	12	1925.70	14	241.58	16	998.00	17	> 12h
		45	135.05	10	143.20	18	22.75	20	23.80	11	272.06
		20	16.31	1	34.01	3	2.64	2	62.51	4	> 12h
15		25	7.86	1	9.74	1	5.53	1	4.28	1	> 12h
		45	475.65	8	246.80	10	305.51	3	153.75	8	> 12h

Tabela 4.35: Tempo de CPU (seg) e $\#cortes$ - 50 localizações.

Também no grupo de maior dimensão, a denominada 1ª estratégia alcança resultados claramente melhores. O que se verifica neste grupo é que a diferença relativamente à 2ª estratégia e ao corte reforçado heuristicamente (secção 4.3) não é tão acentuada. A tabela 4.37 apresenta o resumo em termos de valores médios para o conjunto de instâncias com 50 localizações.

	tempo	cortes
corte usual	3636.56	17.4
corte fortalecido	710.76	8.7
reforço Heur.	279.93	8.2
1ª Estratégia	101.05	9.7
2ª Estratégia	189.33	8.4
Cplex	3364.45	-

Tabela 4.36: Valores médios - 20 localizações.

Naturalmente que o aumento do número de períodos de tempo no horizonte de planeamento conduz, em geral, a tempos de *CPU* superiores. O mesmo sucede quando se aumenta o número de localizações. Tal não é de admirar uma vez que faz aumentar o número de variáveis inteiras no modelo fazendo com que o tempo de resolução pelo procedimento de *Branch-and-Bound* do *IlogCplex 9.0* aumente consideralvelmente, pelo consequente aumento do número de nós de pesquisa. No entanto verificou-se ao longo dos resultados que este tem uma boa *performance* na resolução de instâncias de pequena dimensão. A tabela seguinte apresenta o tempo total consumido quer pela melhor estratégia de corte (que se concluiu ser a estratégia apresentada na subsecção 4.5.3) e o procedimento de *Branch-and-Bound* do *IlogCplex 9.0* [13], considerando todas as instâncias com 5, 10 e 15 períodos de tempo.

	$T = 5$	$T = 10$	$T = 15$
Cplex	40.54	> 60530.05	> 227353.04
1ª Estratégia	9.81	612.55	3262.93

Tabela 4.37: Tempo de *CPU* total.

Como se pode observar pela tabela 4.37, nem sequer para o conjunto de instâncias em que se considerou um horizonte de planeamento dividido em apenas 5 períodos de tempo, o *general solver* conseguiu melhor tempo do que a 1ª estratégia de reforço com base no corte da formulação $PLMC_3$. Porém há pontualmente situações em que o *general solver* conduz a tempos de *CPU* inferiores mas quando tal acontece, essa diferença é muito ligeira.

Em síntese conclui-se que, para a resolução deste conjunto de 54 instâncias, o método da decomposição de Benders mostrou-se bastante eficiente quando se combina a variante de inicialização 2 e a introdução, em cada iteração, de cortes construídos com base na formulação $PLMC_3$ e reforçados heuristicamente através do algoritmo 4.4.

Capítulo 5

Conclusões

Ao longo deste trabalho estudou-se um problema de *phase-in/phase-out* multi-periódico para localização de serviços envolvendo restrições de capacidade.

No capítulo 1 fez-se uma breve contextualização do problema. No capítulo 2 foi feita uma síntese de alguns trabalhos da literatura que têm como objectivo de estudo problemas de localização (ou neles inserido) que têm em conta variações de parâmetros ao longo do tempo.

No capítulo 3 foram apresentados 4 modelos em programação linear inteira mista cuja *performance* foi avaliada com base em testes computacionais usando um conjunto de instâncias geradas aleatoriamente. As 54 instâncias utilizadas foram geradas de forma a variar o número de localizações, o número de clientes e o número de períodos que constituem o horizonte de planeamento, fazendo variar, ainda, o número de serviços em actividade no início do horizonte de planeamento (página 20). Foi utilizado o procedimento de *Branch-and-Bound* do *IlogCplex 9.0* [13] usando a linguagem de modelação *IlogConcert 2.0* [12]. A utilização do *general solver* revelou-se ineficiente nas instâncias de média e grande dimensões testadas, e nem mesmo a utilização de um *upper-cut-off* inicial evitou que para algumas delas não se obtivesse a solução óptima em tempo inferior a 12 horas. No entanto, o procedimento de *Branch-and-Bound* do *general solver*, mostrou uma boa *performance* na resolução de instâncias de menor dimensão.

No que se refere a cada uma das formulções e tendo em consideração os tempos de CPU obtidos, a formulação $PLMC_3$ foi a que apresentou melhores resultados, nomeadamente nas instâncias de maior dimensão. No entanto, os limites inferiores obtidos a partir da relaxação linear são de pior qualidade que os limites inferiores obtidos a partir da relaxação linear da formulção $PLMC_4$. Pôde concluir-se ainda, que a introdução do conjunto de desigualdades válidas (3.16) (página 19) em cada uma das formulações $PLMC_1$ e $PLMC_2$ (dando origem, respectivamente, a $PLMC_3$ e $PLMC_4$), permitiu reduzir, na maioria dos casos significativamente, os tempos de CPU, embora os valores das respectivas relaxações em programação linear se tenham mantido inalteradas.

Ainda no capítulo 3, foi apresentado um algoritmo heurístico de forma a obter um limite superior para o problema. Considerando o mesmo conjunto de instâncias acima referidas, a experiência computacional permitiu considerar as soluções admissíveis obtidas, de boa qualidade, tendo em conta os desvios obtidos (relativamente ao valor óptimo), e que se mantiveram mais ou menos estáveis, mesmo nas instâncias de maior dimensão. A solução dada pela heurística serviu de base a uma das variantes de inicialização no processo de decomposição estudado no capítulo 4.

O capítulo 4 foi dedicado à aplicação de um método de decomposição na tentativa de resolver as instâncias de maior dimensão de forma mais eficiente. Para tal foram tentadas várias estratégias de construção dos cortes a introduzir no problema mestre relaxado. Os chamados 'cortes usuais' provenientes quer da formulação $PLMC_3$ quer da formulação $PLMC_4$, conduzem a tempos de *CPU* muito elevados (principalmente para as instâncias com 15 períodos de tempo), pois têm que ser gerados em grande número para se obter a convergência.

Usou-se então uma forma simples para reforçar o corte resolvendo pequenos problemas de programação linear. Procurou-se, desta forma, alterar coeficientes de algumas variáveis. Os resultados melhoraram: o tempo de *CPU* decresceu assim como o número de cortes gerados e, em alguns casos, estas diminuições foram muito consideráveis. Mesmo assim, os tempos obtidos podem ainda considerar-se elevados, especialmente nas instâncias que consideram o horizonte de planeamento dividido em 15 períodos de tempo. Com o aumento do número de períodos e o eventual aumento do número de problemas de programação linear necessários para o fortalecimento do corte, os tempos podem aumentar significativamente. De forma a tentar ultrapassar essa dificuldade propôs-se um procedimento heurístico para o reforço de um corte (página 84). Assim, para cada par (i, t), em vez da resolução de um problema de programação linear, é aplicado um procedimento heurístico com o qual se tenta diminuir um coeficiente decrementando a variável v_{it} associada ao par (i, t), à custa do aumento de algumas variáveis π_{ijt}, $j \in J$. Esta substituição permitiu melhorar consideravelmente os resultados. Tentou-se ainda a aplicação do mesmo procedimento heurístico de reforço no fortalecimento do corte usual que deriva da formulação $PLMC_3$. Verificaram-se melhorias significativas no número de cortes gerados e, principalmente, nos tempos de *CPU*.

Uma segunda estratégia usando o reforço heurístico no corte usual da formulação $PLMC_3$ foi tentada. Não sendo de desprezar, não revelou a mesma capacidadae que a primeira estratégia.

Apesar de ainda se revelarem algumas dificuldades, podemos concluir que o algoritmo da decomposição de Benders usando cortes gerados com base na formulação $PLMC_3$ e reforçados heuristicamente de acordo com a estratégia apresentada na subsecção 4.5.2 revela-se uma boa alternativa à utilização do procedimento de *Branch-and-Bound* do *IlogCplex 9.0* [13] na resolução do problema estudado.

A *performance* do algoritmo de Benders não depende unicamente, mas também, da qualidade dos cortes a introduzir em cada iteração. A resolução do problema mestre relaxado, tratando-se de um problema de programação linear inteira mista, é uma dificuldade que não pode ser ignorada. Neste trabalho foi usado na sua resolução o procedimento de *Branch-and-Bound* do *IlogCplex 9.0* [13] que revelou em muitas situações enormes dificuldades, conduzindo a tempos de resolução demasiado elevados.

No trabalho de Devine e McDaniel [36], para a resolução do problema mestre, propõem primeiramente a resolução deste relaxando as variáveis inteiras, tratando-o como um problema de programação linear nas primeiras K iterações (sugerem como exemplo K=10) gerando K cortes que são naturalmente válidos para o problema inteiro. Só depois introduzem as restrições de integralidade. Mas também sugerem que podem ser gerados continuamente até não ser possível mais iterações. Nesse caso são gerados os K cortes necessários para a resolução da relaxação em programação linear do problema mestre relaxado.

Cordeau *et al.* [11], propõem uma estratégia semelhante: sugerem relaxar as variáveis inteiras no problema mestre e gerar cortes até se verificar $\frac{LS-LI}{LI} < 0.001$, introduzindo em seguida as restrições de integralidade e resolvendo o problema mestre até alcançar uma solução com 1 % de optimalidade. Estes referem que raramente se justifica a resolução até à optimalidade, uma vez que os dados usados no mundo real, como custos, procura e estimação das capacidades, contêm uma margem de erro superior a 1 %.

O facto de se relaxarem as restrições de integralidade, produzindo soluções não inteiras, obriga a que se tenham que introduzir cortes de admissibilidade, podendo perder-se a admissibilidade do subproblema primal e consequentemente o subproblema dual poder ser ilimitado.

Os resultados positivos aqui apresentados sugerem que ainda poderão ocorrer melhorias se se tiver especial atenção na resolução do problema mestre. Van Roy [49], assim como Magnanti e Wong [34] propõem uma forma de reduzir o número de problemas mestre a resolver usando uma estratégia de decomposição que explora quer a estrutura primal quer a estrutura dual do problema: a técnica da decomposição cruzada. A aplicação deste método para este problema é também uma direcção possível e conveniente assim como a implementação de um algoritmo de *Branch-and-Bound* de forma a tornar o processo independente do *general solver*. Efectuar testes computacionais em instâncias de dimensão superior às utilizadas no presente trabalho permitirá fazer uma avaliação em sistemas de facto de larga escala.

Uma generalização interessante do problema *phase-in/phase-out* aqui apresentado seria considerar as capacidades dos serviços variáveis ao longo do horizonte de planeamento mas consideradas como variáveis de decisão do problema. Shulman [47] sublinha no seu artigo o facto de muito pouca atenção ser dada a problemas que tenham em conta esta variante. Outras variantes passariam por incorporar outro tipo de restrições tais como restrições orçamentais.

Bibliografia

[1] N. Alguacil and A. J. Conejo. Multiperiod optimal power flow using benders decomposition. *IEEE Transactions on Power Systems*, 15:196–201, 2000.

[2] M. J. Alves, J. N. Clímaco, and C. H. Antunes. *Programação linear multiobjectivo: Do modelo de programação linear clássica à consideração explícita de várias funcões objectivo.* Imprensa da Universidade de Coimbra, 2003.

[3] R. H. Ballou. Dynamic warehouse location analysis. *Journal of Marketing Research*, pages 271–276, 1968.

[4] J. E. Beasley. Langrangean heuristics for location problems. *European Journal of Operational Research*, 65:383–399, 1993.

[5] A. Benchakroun, J. A. Ferland, and V. Gascon. Benders decomposition for network design problems with underlying tree structure. *Investigación Operativa*, 6:165–180, 1998.

[6] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[7] T. Benoist, E. Gaudin, and B. Rottembourg. *Constraint programming contribution to Benders decomposition: a case study.* http://www.e-lab.bouygues.fr/ADHOC/publications/, 2002.

[8] M. Carey and A. Srinivasan. Solving a class of network models for dynamic flow control. *European Journal of Operational Research*, 75:151–170, 1994.

[9] P. Chardaire, A. Sutter, and M.-C. Costa. Solving the dynamic facility location problem. *Networks*, 28:117–124, 1996.

[10] N. Christofides and J. E. Beasley. A tree search algorithm for the p-median problem. *European Journal of Operational Research*, 10:196–204, 1982.

[11] J. F. Cordeau, F. Passin, and M. M. Solomon. An integrated model for logistics network design. *Les Cahiers du GERARD, G-2002-07*, 2002.

[12] CPLEX. *ILOG CONCERT User's Manual*, 2004.

[13] CPLEX. *ILOG CPLEX User's Manual*, 2004.

[14] J. Dias, M. E. Captivo, and J. Clímaco. Problema de localização dinâmica com abertura, fecho e reabertura de serviços: formulação e heurística primal-dual. Technical report, 6, INESC-Coimbra, 2002.

[15] J. Dias, M. E. Captivo, and J. Clímaco. Capacitated dynamic location problems with opening, closure and reopening of facilities. Technical report, 2, INESC-Coimbra, 2004.

[16] D. Erlenkotter. A dual-based procedure for uncapacitated facility location. *Operations Research*, 26:992–1009, 1978.

[17] D. Erlenkotter. A comparative study of approaches to dynamic location problems. *European Journal of Operational Research*, 6:133–143, 1981.

[18] C. Ferreira, M. E. Captivo, J. Clímaco, and B. S. Santos. Modelos multiobjectivo para a instalação de equipamentos desagradáveis. Technical report, 13, CIO - Centro de Investigação Operacional, 1995.

[19] R. D. Galvão. Uncapacitated facility location problems: contributions. *Pesquisa Operacional*, 24:7–38, 2004.

[20] R. D. Galvão and E. del R. Santibañez-Gonzalez. The p_k-median dynamic location problem formulation and a heuristic solution method. *Investigación Operativa*, 1:295–308, 1990.

[21] R. D. Galvão and E. del R. Santibañez-Gonzalez. A lagrangean heuristic for the p_k-median dynamic location. *European Journal of Operational Research*, 58:250–262, 1992.

[22] L. Gao and E. Powell Robinson Jr. Uncapacitated facility location: general solution procedure and computational experience. *European Journal of Operational Research*, 76:410–427, 1994.

[23] Saul Gass. *Linear programming: methods and applications*. McGraw-Hill book company, Third edition, 1969.

[24] A. Geoffrion and W. Graves. Multicommodity distribution system design by benders decomposition. *Management Science*, 20:822–844, 1974.

[25] R. Gomes and B. Nunes. *Projecto de Investigação Operacional 2002/2003*. Departamento de Estatística e Investigação Operacional. FCUL.

[26] O. Goussevskaia and G. R. Mateus. Problema de localização de serviços e servidores em sistemas de comunicação móvel de terceira geração: Modelo e algoritmo. *Simpósio Brasileiro de Pesquisa Operacional*, 2001.

[27] M. Guignard. A langrangean dual ascent algorithm for simple plant location problems. *European Journal of Operational Research*, 35:193–200, 1988.

[28] M. Guignard and K. Spielberg. A direct dual method for the mixed plant location problema with some side constraints. *Mathematical Programming*, 17:198–228, 1979.

[29] S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12:450–459, 1964.

[30] Y. Hinojosa, J. Puerto, and F. R. Fernández. A multiperiod two-echelon multicommodity capacitated plant location problem. *European Journal of Operational Research*, 123:271–291, 2000.

[31] B. F. Hobbs, Y. Ji, and S. A. McCusker. Distributed utility planning usind probabilistic production costing and generalized benders decomposition. *IEEE Transactions on Power Systems*, 17:497–505, 2002.

[32] M. Labbé, D. Peeters, and J. F. Thisse. Location on networks. In M. O. Ball et al., editor, *Handbooks in OR and MS*, 1995.

[33] J. M. Y. Leung and T. L. Magnanti. Valid inequalities and facets of the capacitated plant location problem. *OR*, 1986.

[34] T. L. Magnanti and R. T. Wong. Decomposition methods for facility location problems. In P. B. Mirchandani and R. L. Francis, editors, *Discrete Location Theory*, 1990.

[35] T. L. Magnanti and R. T. Wong. Accelerating benders decomposition: algorithmic enhancement and model selection criteria. *European Journal of Operational Research*, 29:464–483, 1991.

[36] D. McDaniel and M. Devine. A modified benders partitioning algorithm for mixed integer programming. *Management Science*, 24:312–319, 1977.

[37] E. Melachrinoudis and H. Min. The dynamic relocation and phase-out of a hybrid, two-echelon plant/warehousing facility: A multiple objective approach. *European Journal of Operational Research*, 23:1–15, 2000.

[38] M. T. Melo, S. Nickel, and F. Saldanha da Gama. Dynamic multi-commodity capacitated facility location: A mathematical modeling framework for strategic supply chain planning. *Computers and Operations Research (to appear).*

[39] M. T. Melo, S. Nickel, and F. Saldanha da Gama. Large-scale models for dynamic multi-commodity capacitated facility location. Technical report, 58, Fraunhofer Institute for Industrial Mathematics (ITWM), Kaiserslautern, Germany (disponível em http://www.itwm.fhg.de/), 2003.

[40] M. T. Melo and R. Vélasquez. Solving a large-scale dynamic facility location problem with variable neighbourhood and token ring search. *Apresentado no 39º encontro da Sociedade Neo-Zelandeza de Investigação Operacional, Auckland, Nova Zelândia*, 28.

[41] S. H. Owen and M. S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111:423–447, 1998.

[42] M. C. Rafael, M. T. Schilling, and E. L. Silva. Modelo integrado geração-transmissão para programação de desligamentos de geradores no planejamento da operação. *SBA Controle e Automação*, 9:127–134, 1998.

[43] C. D. Randazzo, H. P. Luna, and P. Mahey. Benders decomposition for local access network design with two technologies. *Discrete Mathematics and Theoretical Computer Science*, 4:235–246, 2001.

[44] G. M. Roodman and L. B. Schwarz. Optimal and heuristic facility phase-out strategies. *AIIE Transactions*, 7:177–184, 1975.

[45] G. M. Roodman and L. B. Schwarz. Extensions of the multi-period facility phase-out model: new procedures and application to a phase-in/phase-out problem. *AIIE Transactions*, 9:103–107, 1977.

[46] F. Saldanha da Gama. *Modelos e algoritmos para o problema de localização dinâmica*. Universidade de Lisboa, 2002. Dissertação de doutoramento.

[47] A. Shulman. An algorithm for solving dynamic capacitated plant location problems with discrete expansion sizes. *Operations Research*, 39:423–436, 1990.

[48] M. M. Syslo, N. Deo, and J. Kowalik. *Discrete Optimization Algorithms*. Prenctice-Hall, 1983.

[49] T. Van Roy. A cross decomposition algorithm for capacitated facility location. *Operations Research*, 34:145–163, 1986.

[50] T. Van Roy and D. Erlenkotter. A dual-based procedure for dynamic facility location. *Management Science*, 28:1091–1105, 1982.

[51] G. O. Wesolowsky and W. G. Truscott. The multiperiod location-allocation problem with relocation of facilities. *Management Science*, 22:57–65, 1975.

Apêndice

Apresentar-se-ão neste apêndice alguns dos resultados que por algum motivo não foram exibidos ao longo do trabalho mas que não podem deixar de ser registados. Seguidamente apresentam-se os resultados da aplicação da decomposição de Benders o corte usual tendo por base a formulação $PLMC_3$. A tabela $A1$ apresenta os resultados para as instâncias com 10 localizações e a tabela $A2$ resume os valores médios.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		3	0.08	1	0.09	2	0.06	1
5		5	0.39	5	0.26	5	0.36	5
		8	0.27	3	0.22	4	0.20	3
		3	1.02	6	0.87	6	0.80	5
10	20	5	1.53	6	0.82	5	0.90	5
		8	1.35	4	1.00	5	1.30	4
		3	469.00	13	328.99	15	463.83	15
15		5	17648.62	32	12360.61	31	17071.94	29
		8	9.74	8	12.78	8	20.97	12
		3	1.00	2	0.13	2	0.20	2
5		5	1.28	3	0.37	4	0.30	3
		8	1.84	6	0.62	6	0.67	7
		3	8.13	5	3.66	5	4.38	4
10	50	5	4.22	4	0.64	4	0.63	4
		8	13.78	15	21.42	18	14.78	16
		3	329.86	22	347.86	20	331.67	18
15		5	193.08	12	246.49	13	235.26	13
		8	1238.01	28	998.28	26	1406.94	29
		3	2.83	4	0.36	4	0.42	4
5		5	2.83	1	0.27	3	0.25	2
		8	3.42	5	0.50	5	0.58	5
		3	13.69	9	3.72	9	3.64	9
10	100	5	16.05	10	2.61	6	6.95	10
		8	10.81	6	1.38	6	1.75	7
		3	83.27	9	168.78	9	94.89	9
15		5	1509.01	24	1119.90	22	1549.83	26
		8	115.21	13	110.03	12	71.61	14

Tabela A1: Tempo de *CPU* (seg) e *#cortes* - 10 localizações

variante	tempo	cortes
Inicialização 1	802.96	9.5
Inicialização 2	582.69	9.4
Inicialização 3	788.34	9.7

Tabela A2: Valores médios - 10 localizações

Tendo em conta as tabelas acima, pode verificar-se que apesar de o corte ter como base a formulação $PLMC_3$, produz elevados tempos de CPU, assim como um elevado número de cortes. Comparando com a *performance* do corte usual de Benders que tem por base a formulação $PLMC_4$ e tendo em conta as tabelas $A2$ e 4.4 (página 68), verifica-se que ao nível do número médio de cortes é muito equilibrado mas em relação aos tempos médios de CPU há algumas diferenças, dependendo da variante com que foi inicializado o processo. Em relação à variante de inicialização 1, embora ambas as estratégias de corte tenham gerado igual número médio de cortes na resolução do conjunto de instâncias com 10 localizações, usando o corte usual de $PLMC_3$ aqui apresentado, conduziu a um tempo de CPU mais elevado. O mesmo acontece com a inicialização 3 embora a diferença de tempos seja menos substancial. Relativamente à inicialização 2, a utilização do corte usual da formulação $PLMC_3$ coduziu a tempo médio inferior ao corte usual da secção 4.2, apesar deste último ter gerado um menor número médio de cortes.

Este corte só funcionou bem em instâncias que consideram o horizonte de planeamento dividido em 5 e 10 períodos de tempo, independentemente do número de clientes. Os resultados referentes às instâncias com 20 e 50 localizações constam nas tabelas $A3$ e $A4$, respectivamente .

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	#J	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		8	9.73	2	0.27	3	0.25	2
5		10	7.42	2	0.30	3	0.22	2
		16	15.23	25	5.87	33	4.72	29
		8	315.91	21	488.28	23	576.00	19
10	50	10	1059.32	29	1005.89	36	1477.95	31
		16	207.08	15	219.22	17	126.76	21
		8	> 12h	-	> 12h	-	> 12h	-
15		10	28849.72	62	> 12h	-	22538.81	53
		16	5156.78	27	3747.06	25	4306.00	33
		8	8.92	5	0.61	5	0.91	6
5		10	7.83	3	0.33	3	0.31	2
		16	14.25	25	4.99	30	4.26	25
		8	278.62	28	259.93	26	208.05	29
10	100	10	76.11	23	*out mem.*	-	*out mem.*	-
		16	304.70	35	395.30	41	339.86	44
		8	524.53	13	8284.14	34	5582.19	31
15		10	> 12h	-	> 12h	-	> 12h	-
		16	3304.77	21	5339.43	20	4496.39	20

Tabela A3: Tempo de *CPU* (seg) e *#cortes* - 20 localizações

⌜ *Para a instância assinalada com 'out mem.' na tabela A3, o solver apresentou um erro de memória aquando da resolução de* PM_R. ⌟

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	#J	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		20	85.44	5	2.02	5	4.38	4
5		25	94.84	6	1.84	6	1.86	6
		45	115.61	24	13.08	29	14.61	25
		20	3253.75	26	2046.48	32	2648.08	27
10	100	25	11098.99	25	> 12h	-	> 12h	-
		45	707.11	25	62.67	24	56.59	22
		20	1005.74	4	112.24	4	221.00	3
15		25	*out mem.*	-	4.80	1	*out mem.*	-
		45	1913.36	21	1584.96	19	1776.73	17

Tabela A4: Tempo de *CPU* (seg) e *#cortes* - 50 localizações

⌜ *Para a instância assinalada com 'out mem.' na tabela A4, o solver apresentou um erro de memória aquando da resolução de* PM_R. ⌟

Para o grupo de instâncias com 20 e 50 localizações, continuam a evidenciar-se as dificuldades em obter uma solução óptima com este corte. Tal como aconteceu com o corte usual da formulação $PLMC_4$, o número de cortes que é necessário gerar é muito elevado e há, inclusivamente, mais instâncias para as quais não é encontrada a solução óptima em tempo inferior de 12 horas, do que quando usado o corte da secção 4.2. No entanto há também muitas instâncias em que, usando o corte com base em $PLMC_3$, são produzidos tempos de CPU inferiores, nomeadamente, nas instâncias de menor dimensão (com 5 e 10 períodos de tempo).

À semelhança do que foi feito com o corte usual da formulação $PLMC_4$, também neste caso podemos tentar fortalecer o corte usual resolvendo o seguinte problema de programação linear:

(PR'_{it})

$$\begin{aligned} &\min_{v_{it} \geq 0} \quad v_{it}\, Q_i \\ &s.\,a: \\ &\quad \lambda_{jt} - d_{jt}\, v_{it} \leq c_{ijt} \qquad\qquad j \in J \end{aligned}$$

Sendo resolvido para cada par (i, t) nas condições do procedimento 4.2, substituindo-se a resolução do problema (4.24) pelo problema acima. O algoritmo da decomposição mantém-se tal como o algoritmo 4.3. Em seguida apresentam-se os resultados obtidos para o conjunto de instâncias que consideram 10 localizações.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		3	0.58	1	0.53	1	0.56	1
5		5	1.19	3	1.14	3	1.19	3
		8	0.75	2	0.67	2	0.70	2
		3	3.83	3	3.70	3	3.98	3
10	20	5	4.53	4	4.14	4	3.79	4
		8	1.96	2	1.80	2	1.84	2
		3	8.20	3	7.70	3	30.43	6
15		5	753.56	8	1127.69	8	1135.58	8
		8	6.31	2	2.95	2	2.84	2
		3	1.42	1	0.80	1	0.62	1
5		5	2.45	3	1.49	3	1.05	2
		8	3.16	5	1.86	5	1.95	5
		3	6.45	2	2.67	2	2.69	2
10	50	5	5.58	2	1.95	2	1.95	2
		8	10.19	5	6.66	5	9.48	7
		3	60.35	7	46.95	7	27.16	7
15		5	34.03	4	21.92	4	19.38	4
		8	113.17	10	101.59	10	89.50	10
		3	4.22	2	1.44	2	1.34	2
5		5	3.36	1	0.62	1	1.25	2
		8	3.42	1	0.53	1	0.53	1
		3	14.40	3	4.23	3	4.49	3
10	100	5	10.28	2	2.22	2	2.27	2
		8	10.19	1	1.16	1	1.11	1
		3	32.15	3	6.37	3	8.37	4
15		5	49.33	5	28.00	5	31.50	5
		8	42.11	3	6.52	3	6.35	3

Tabela A5: Tempo de *CPU* (seg) e *#cortes* - 10 localizações

Variante	tempo	cortes
Inicialização 1	43.97	3.3
Inicialização 2	51.38	3.3
Inicialização 3	51.55	3.5

Tabela A6: Valores médios - 10 localizações

Observando a tabela $A6$ rapidamente se conclui que o fortalecimento do corte usual permite reduzir drasticamente quer os tempos de CPU quer o número de cortes necessários para obter a solução óptima do problema, relativamente ao corte usual. Comparando com os resultados obtidos com o fortalecimento do corte usual da formulação forte (página 82), podemos constatar que são muito semelhantes, quer no tempo produzido quer no número de cortes gerados. Vejamos a seguir para as instâncias com 20 e 50 localizações.

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		8	10.59	1	1.05	1	1.03	1
5		10	8.14	1	0.95	1	0.98	1
		16	20.84	1	9.56	10	12.45	13
		8	57.53	5	21.08	5	48.16	7
10	50	10	157.00	8	254.63	9	267.31	13
		16	102.16	7	46.95	7	46.58	8
		8	4081.44	11	6732.86	11	5192.00	11
15		10	3935.24	18	4460.48	18	2889.93	17
		16	1458.72	12	1231.94	11	1506.17	10
		8	35.22	4	4.56	4	4.58	4
5		10	28.36	2	2.05	2	2.06	2
		16	30.38	8	8.36	8	10.28	10
		8	135.19	7	25.56	7	17.56	6
10	100	10	161.58	8	22.27	8	21.73	8
		16	140.99	6	41.04	8	50.28	10
		8	442.72	6	123.81	6	98.75	6
15		10	18473.69	8	$>$ 12h	-	33207.77	10
		16	697.44	6	494.86	6	98.49	5

Tabela A7: Tempo de CPU (seg) e *#cortes* - 50 localizações

			c/ Sol. Heur		c/ Serv. abertos		s/ cortes inic.	
T	$\#J$	$\#I^c$	*tempo*	*#cortes*	*tempo*	*#cortes*	*tempo*	*#cortes*
		20	89.13	2	5.38	2	5.37	2
5		25	100.33	3	7.46	3	10.19	4
		45	146.83	18	31.25	14	28.45	13
		20	6550.88	14	942.00	16	7412.88	16
10	100	25	1188.48	11	1650.94	10	5572.50	12
		45	751.97	15	104.14	14	109.30	15
		20	866.81	4	30.42	2	25.42	2
15		25	*out mem.*	-	4.58	1	*out mem.*	-
		45	1849.79	7	299.97	8	876.97	9

Tabela A8: Tempo de *CPU* (seg) e *#cortes* - 50 localizações

⌜ *Para a instância assinalada com 'out mem.' na tabela A8, o solver apresentou um erro de memória aquando da resolução de PM_R.* ⌟

Na maioria das instâncias destes dois últimos conjuntos, a variante de inicialização 2 apresenta tempos de *CPU* inferiores aos tempos das restantes variantes de inicialização. Também o número de cortes gerados é, na maioria dos casos, igual ou inferior.

Comparativamente, os tempos que aqui se apresentam tendem a ser inferiores aos tempos obtidos usando o corte usual apresentado na secção 4.2. No grupo de instâncias com 20 localizações e considerando 5 e 10 períodos de tempo, o corte usual da secção 4.2 apresenta, com uma excepção, tempos superiores aos tempos aqui apresentados (considerando as três variantes de inicialização). O mesmo sucedendo com as instâncias do conjunto com 50 localizações, sendo neste caso todas as instâncias com 5 períodos de tempo. Em relação aos tempos para as instâncias que consideram 15 períodos no horizonte de planeamento, são, em geral, mais baixos usando o corte de $PLMC_4$.

Em jeito de conclusão, pode dizer-se que qualquer dos cortes usuais, de $PLMC_3$ ou $PLMC_4$ não constituem boas escolhas para o algoritmo da decomposição de Benders.