Emparelhamentos, Casamentos Estáveis e Algoritmos de Colocação de Professores

Ana Paula Tomás

Technical Report Series: DCC-05-02

Departamento de Ciência de Computadores – Faculdade de Ciências

&

Laboratório de Inteligência Artificial e Ciência de Computadores

---

Universidade do Porto

Rua do Campo Alegre, 823, 4150 Porto, Portugal

Tel: +351+22+6078830 – Fax: +351+22+6003654

`http://www.dcc.fc.up.pt/Pubs/treports.html`

Emparelhamentos, Casamentos Estáveis e Algoritmos de Colocação de Professores

Ana Paula Tomás*
DCC-FC & LIACC, Universidade do Porto
R. do Campo Alegre, 823
4150-180 Porto, Portugal
apt@ncc.up.pt

Março 2005 (revisto Maio 2005)

Resumo

Este relatório apresenta os resultados dum estudo sobre problemas de afectação com preferências, em que se analisou, em particular, algoritmos para resolução do problema da colocação de educadores e professores por concurso nacional, no contexto da actual legislação portuguesa. Estabelece a relação entre esse problema de colocação e variantes dum problema clássico em optimização combinatória, designado por problema dos casamentos estáveis. Tal permitiu deduzir algumas propriedades interessantes e importantes das listas de colocações admissíveis. Mostra que não se pode pressupor que as listas de preferências dos candidatos estão totalmente ordenadas sem se perder a garantia de obtenção de listas de colocações justas, as quais são designadas por listas de colocações óptimas segundo os candidatos. Define exactamente este conceito em termos matemáticos, o qual, em cada fase do concurso, traduz a atribuição a cada candidato da melhor posição, sem prejuízo da observância do mesmo para todos os que o precedam na lista ordenada de candidatos. Prova a existência de algoritmos polinomiais para a determinação dessas listas óptimas. Considerando a dimensão das instâncias reais deste problema, propõe métodos alternativos para a sua resolução, que, podendo não ser polinomiais, podem contudo ter na prática melhor desempenho. Embora não tenha sido acompanhado duma análise experimental que melhor o pudesse suportar, face à complexidade das instâncias reais, conjectura a necessidade de, a curto ou médio prazo, introduzir alterações à lei, de forma a garantir que o problema da determinação de listas óptimas (justas) se possa resolver em tempo útil.

1 Introdução

Este trabalho teve por motivação a polémica gerada em torno do *Concurso de Educadores de Infância e de Professores dos Ensinos Básico e Secundário*, relativo ao ano escolar de 2004/05, em Portugal, amplamente veiculada pelos orgãos de comunicação social, nos meses de Setembro e Outubro de 2004. Surge do reconhecimento de que poderia ser estabelecida alguma relação entre o problema da elaboração de listas de colocações e o designado por problema dos casamentos estáveis – StableMarriage [9].

*Trabalho parcialmente financiado pelo LIACC através do *Programa de Financiamento Plurianual da Fundação para a Ciência e Tecnologia* e do *Programa POSI.*

Na sequência disso, no primeiro semestre do ano lectivo de 2004/05, foi proposto um mini-projecto prático sobre o tema[1], aos alunos das disciplinas de "Métodos de Apoio à Decisão" e de "Investigação Operacional" do 4º ano das licenciaturas em Ciência de Computadores e em Engenharia de Redes e Sistemas Informáticos do Departamento de Ciência de Computadores da Faculdade de Ciências da Universidade do Porto e também da disciplina de "Métodos Quantitativos e Restrições" do mestrado em Informática do mesmo departamento. Tinha por objectivo o estudo de algoritmos para determinação de listas de colocações, esperando-se dos alunos que tratassem pelo menos uma situação mais simples do que a real – aquela em que se assumia que as listas de preferências dos docentes estavam totalmente ordenadas – ou seja, que não existiam posições pelas quais os docentes tinham manifestado igual preferência. Alguns alunos dão conta de que a solução desenvolvida pela ATX Software[2] tinha sido por esta divulgada e estava disponível na Web [2, 8], e acabam mesmo por implementar essa proposta nos seus mini-projectos.

O presente relatório resulta dum estudo conduzido depois duma análise cuidada da solução que a ATX Software divulgou. Para sua elaboração foi necessário proceder a uma leitura com alguma profundidade do Decreto-Lei nº 35/2003, de 27 de Fevereiro, que regula o referido concurso, e ainda a um trabalho de pesquisa sobre o problema de StableMarriage e suas variantes.

O relatório começa por apresentar alguns problemas de casamentos estáveis e algoritmos existentes para sua resolução, bem como resultados que estabelecem que algumas variantes podem ser NP-hard. Analisa o algoritmo da ATX Software e mostra que o mesmo pode ser melhorado, usando a relação do problema de colocação com o de casamentos estáveis. Apresenta alguns exemplos hipotéticos (mas que fazem sentido no contexto da actual legislação) em que, ao assumir-se o pressuposto de que as listas de preferências dos candidatos estão totalmente ordenadas (como se tem feito e o faz a solução divulgada pela ATX Software), se podem introduzir injustiças nas colocações nalgumas fases do concurso. Propõe modelos matemáticos alternativos e estratégias para obtenção de listas de colocações justas, as quais são designadas por *listas de colocações óptimas (segundo os candidatos)*. É importante salientar que estas listas óptimas não violam o consignado no referido Decreto-Lei, podendo não ser exactamente, as que melhor satisfazem os candidatos. Mostra que existem algoritmos que permitem determinar listas de colocações óptimas em tempo polinomial, mas atendendo à dimensão das instâncias reais aponta para uma possível necessidade de a curto ou médio prazo introduzir alterações à lei.

2 O Problema dos Casamentos Estáveis

O problema dos Casamentos Estáveis (*Stable Marriage Problem*) e suas variantes tem sido objecto de estudo em várias publicações científicas, dado o seu interesse prático. A versão clássica do problema traduz-se da forma seguinte.

> StableMarriage:
>
> Supondo que cada elemento dum grupo de n homens e n mulheres ordenou todos os de sexo oposto por ordem de preferência estrita, pretende-se determinar um emparelhamento estável.

[1] `http://www.dcc.fc.up.pt/~apt/aulas/MAD/0405/enun2004_alternativo.html`

[2] Empresa que desbloqueou a situação resolvendo o problema das colocações no final de Setembro de 2004.

Sendo $\mathcal{H} = \{h_1, \ldots, h_n\}$ e $\mathcal{M} = \{m_1, \ldots, m_n\}$ os conjuntos de homens e mulheres, um emparelhamento E é uma qualquer função injectiva de $\mathcal{H}$ em $\mathcal{M}$. Informalmente, um emparelhamento é, neste caso, um conjunto de n casais (monogâmicos e heterosexuais).

Um emparelhamento E diz-se *instável* se e só se existir um par $(h, m) \notin E$ tal que h prefere m à sua parceira em E e m também prefere h ao seu parceiro em E. Caso contrário, diz-se *estável*.

Exemplo 1 *Para a instância seguinte, em que $n = 4$ e as listas de preferências se consideram ordenadas por ordem (estritamente) decrescente da esquerda para a direita,*

$$\begin{array}{llllllllll} h_1: & m_4, & m_2, & m_3, & m_1 & \quad & m_1: & h_4, & h_2, & h_1, & h_3 \\ h_2: & m_2, & m_3, & m_4, & m_1 & & m_2: & h_3, & h_1, & h_4, & h_2 \\ h_3: & m_2, & m_3, & m_1, & m_4 & & m_3: & h_2, & h_3, & h_1, & h_4 \\ h_4: & m_1, & m_3, & m_2, & m_4 & & m_4: & h_3, & h_4, & h_2, & h_1 \end{array}$$

pode-se verificar, através duma simples análise de casos, que

$$\{(h_1, m_4), (h_2, m_3), (h_3, m_2), (h_4, m_1)\}$$

é um emparelhamento estável (que se obtem pelo Algoritmo de Gale-Shapley*).*

Gale e Shapley [9] mostraram que qualquer instância de StableMarriage admite pelo menos uma solução (ou seja, um emparelhamento estável) e que um tal emparelhamento poderia ser obtido por aplicação do algoritmo seguinte.

Algoritmo de Gale-Shapley (1962)

```
Considerar inicialmente que todas as pessoas estão livres.
Enquanto houver algum homem h livre fazer:
    seja m a primeira mulher na lista de h a quem este ainda não se propôs;
    se m estiver livre então
        emparelhar h e m (ficam noivos)
    senão
        se m preferir h ao seu actual noivo h' então
            emparelhar h e m (ficam noivos), voltando h' a estar livre
        senão
            m rejeita h e assim h continua livre.
fim
```

A ideia da prova da correcção do algoritmo é muito simples, recorrendo a redução ao absurdo. Se existisse um par $(h, m) \notin E$ que tornasse o emparelhamento instável, então h teria que preferir m à noiva m' com que ficou e m teria que preferir h ao noivo com que ficou. Mas, se h prefere m a m' então h propôs-se a m antes de se propor a m'. A mulher m só o rejeitaria, na altura ou posteriormente, para manter ou ficar com um noivo que preferisse. De acordo com o algoritmo, nenhuma mulher rejeita um noivo que prefere para se tornar noiva doutro de que gosta menos. Portanto, não pode existir (h, m) nas condições referidas e, consequentemente, E é estável.

Gale e Shapley mostraram ainda que a complexidade deste algoritmo é $O(n^2)$, o que, mais uma vez, resulta de, no pior dos casos, cada homem só faria n propostas, uma a cada mulher.

O emparelhamento obtido pelo ALGORITMO DE GALE-SHAPLEY é *óptimo* para os homens e *péssimo* para as mulheres: qualquer homem fica com a melhor parceira que pode ter em qualquer emparelhamento estável e cada mulher fica com o pior parceiro. Obviamente, a situação reverte-se se passarem a ser as mulheres que se propõem. A versão seguinte do algoritmo, ainda da autoria dos mesmos autores, permitiu reconhecer esta e outras propriedades estruturais das soluções do problema.

EXTENSÃO DO ALGORITMO DE GALE-SHAPLEY (1962)

> Considerar inicialmente que todas as pessoas estão livres.
> Enquanto houver algum homem h livre fazer:
>   seja m a primeira mulher na lista actual de h;
>   se algum homem p estiver noivo de m então
> p passará a estar livre;
> h e m ficam noivos;
>   para cada sucessor h' de h na lista de m
>   retirar o par (h', m) das listas de h' e m
> fim

Sendo E e E' emparelhamentos estáveis, diz-se que E *domina* E' *segundo os homens* (isto é, os homens preferem E a E'), e escreve-se $E \preceq E'$, se e só se qualquer homem que não tenha a mesma parceira em ambos, tem em E uma que prefere à que tem em E'. Esta relação de dominância confere ao conjunto $\mathcal{E}$ dos emparelhamentos estáveis, a estrutura de reticulado distributivo $(\mathcal{E}, \preceq)$. Dados dois quaisquer emparelhamentos estáveis E e E', o emparelhamento $E \wedge E'$, em que cada homem fica com a mulher que prefere dentre as suas parceiras em E e E', é estável. Também é estável o emparelhamento $E \vee E'$, em que cada homem fica com a mulher de que gosta menos dentre as suas parceiras em E e E'. Estes emparelhamentos $E \wedge E'$ e $E \vee E'$ são o ínfimo e o supremo entre E e E', respectivamente. Mostra-se também que E domina E' segundo os homens se e só se E' domina E segundo as mulheres.

O emparelhamento óptimo para os homens, o qual é determinado pelo ALGORITMO DE GALE-SHAPLEY, é o mínimo do reticulado. O máximo é o emparelhamento óptimo para as mulheres, o qual, como já se referiu pode ser determinado pelo mesmo algoritmo, se se trocarem entre si os papeis que nele desempenham os homens e as mulheres. Se estes emparelhamentos coincidirem então a instância admite um único emparelhamento estável.

Estas e outras propriedades dos emparelhamentos estáveis de STABLEMARRIAGE são exploradas, por exemplo, por Gusfield para desenvolver algoritmos eficientes para alguns problemas [12]. Em particular, apresenta métodos para a determinação de todos os pares estáveis[3] com complexidade $O(n^2)$, para a enumeração de todos os emparelhamentos estáveis, com uma complexidade temporal $O(n^2+n|S|)$ e espacial $O(n^2)$ e para a construção em $O(n^2)$ dum emparelhamento estável que minimiza o grau de descontentamento máximo. Para uma melhor apreciação da relevância destes resultados, é de notar que Irving e Leather provaram em [16] que o número $|S|$ de emparelhamentos estáveis duma instância de STABLEMARRIAGE pode ser exponencial em n.

[3]Um par (h, m) é estável se ocorrer nalgum emparelhamento estável.

3 Variantes do Problema dos Casamentos Estáveis

O trabalho de Gale e Shapley teve como principal motivação a resolução do problema de colocação de alunos em cursos universitários nos EUA.

Cada aluno candidata-se a algumas universidades e formula uma lista de preferências ordenadas estritamente. Cada universidade tem um certo número de vagas e ordena também estritamente os seus candidatos, podendo liminarmente não aceitar alguns. Note-se que nem todas as universidades têm a mesma lista de preferências. Pretende-se colocar os alunos de acordo com as preferências mútuas.

Este problema tem por modelo uma variante de StableMarriage que é designada na bibliografia por StableMarriageWithIncompleteLists. As vagas correspondem às mulheres, os alunos aos homens e as listas de preferências são incompletas (existem parceiros inaceitáveis) mas estritamente ordenadas.

Um emparelhamento (maximal) será um conjunto E de pares (h, m) com $h \in \mathcal{H}$ e $m \in \mathcal{M}$, tal que h e m se consideram mutuamente aceitáveis, não existem pares em E que partilhem elementos e não existem pares de $\mathcal{H} \times \mathcal{M}$ que possam ser acrescentados a E.

Uma das propriedades interessantes das suas soluções é a de que para qualquer instância deste problema, os conjuntos de homens e de mulheres podem ser subdivididos em dois grupos: o daqueles que ficam com parceiro em todos os emparelhamentos estáveis e o daqueles que não têm parceiro em nenhum emparelhamento estável. A prova pode ser encontrada por exemplo em [13]. Tendo em conta esta propriedade pode-se concluir que, independentemente do algoritmo que é usado, serão sempre colocados os mesmos alunos e ocupadas as mesmas vagas (não necessariamente pelos mesmos alunos). Na extensão do Algoritmo de Gale-Shapley que resolve este problema, são os alunos que se propõem às universidades, resultando um emparelhamento óptimo do ponto de vista dos alunos e péssimo para as universidades. A versão adaptada do algoritmo é apresentada a seguir, sendo a tradução da que é dada em [13]. Por razões que abaixo se apresentam, refere internos e hospitais em vez de candidatos e universidades.

Algoritmo de Gale-Shapley (orientado por internos)

```
Considerar inicialmente que todos os internos estão livres.
Considerar também que todas as vagas nos hospitais estão livres.
Enquanto existir algum interno r livre cuja lista de preferências é não vazia
    seja h o primeiro hospital na lista de r;
    se h não tiver vagas
        seja r' o pior interno colocado provisoriamente em h;
        r' fica livre (passa a não estar colocado);
    colocar provisoriamente r em h;
    se h ficar sem vagas então
        seja s o pior dos colocados provisoriamente em h;
        para cada sucessor s' de s na lista de h
            remover s' e h das respectivas listas
fim
```

Alguns anos depois da sua publicação [9] em 1962 descobriu-se que um algoritmo essencialmente análogo estava já a ser usado desde 1952 nos EUA, pelo *National Intern Matching Program* (depois designado *National Resident Matching Program*, NRMP), para colocação de estudantes de medicina nos hospitais para realizarem o internato. Também aqui, cada hospital tem uma lista de preferências própria. Neste algoritmo, são, no entanto, os hospitais que se propõem aos candidatos, resultando num emparelhamento óptimo do ponto de vista dos hospitais.

O critério de estabilidade das soluções é reformulado do modo seguinte. Um emparelhamento é *instável* se e só se existir um candidato r e um hospital h tais que h é aceitável para r e r é aceitável para h, o candidato r não ficou colocado ou prefere h ao seu actual hospital e h ficou com vagas por preencher ou h prefere r a pelo menos um dos candidatos com que ficou. Caso contrário, diz-se *estável*. Nesta situação, o conceito de emparelhamento é o mesmo, se se considerar que a atribuição é de candidatos a vagas.

Demonstra-se que é estável o emparelhamento definido pelas colocações provisórias depois da execução do ALGORITMO DE GALE-SHAPLEY (ORIENTADO POR INTERNOS). Nesse emparelhamento, cada interno que ficar colocado fica no melhor hospital em que poderia ser colocado de forma a garantir a estabilidade do emparelhamento. Cada interno que não ficar colocado por este algoritmo, não poderá ficar colocado em nenhum outro emparelhamento estável determinado por outro método. Dada uma qualquer instância do problema NRMP (internos-hospitais, alunos-universidades) se um hospital h não preencher as suas vagas, então qualquer interno que ficar colocado em h no emparelhamento estável óptimo do ponto de vista dos internos, fica também colocado em h em todos os emparelhamentos estáveis. Este último resultado é conhecido como o *Teorema dos Hospitais Rurais*, por serem aqueles que habitualmente ficam com vagas por preencher.

Em resumo, pode-se mostrar que o número de internos colocados em cada hospital é o mesmo em todos os emparelhamentos estáveis. Ficam por colocar exactamente os mesmos candidatos em todos os emparelhamentos estáveis. Qualquer hospital que não preencha as suas vagas num dado emparelhamento estável, fica com exactamente os mesmos internos em todos os emparelhamentos estáveis.

3.1 Listas de preferências não ordenadas estritamente

Os modelos analisados até aqui pressupõem uma ordenação estrita das listas de preferências, de ambos os grupos. Uma ordenação estrita é o que tecnicamente se designa por *uma ordem total*. Face a duas quaisquer possibilidades distintas x e y, o interveniente (homem/mulher, aluno/universidade, interno/hospital) terá que especificar se prefere x ou se prefere y, não podendo ser indiferente, isto é manifestar igual preferência por ambas.

Existem estudos sobre variantes de STABLEMARRIAGE que contemplam a possibilidade dos conjuntos de preferências não estarem totalmente ordenados, mas apenas parcialmente ordenados (por exemplo, [14, 19]). Um caso particular ocorre quando a ordem é quase uma ordem total, mas existem indiferenças que se traduzem por empates. Duas dessas variantes são STABLEMARRIAGEWITHTIES e STABLEMARRIAGEWITHTIESANDINCOMPLETELISTS. Na primeira supõe-se que as listas de preferências estão completas. Isto é, que qualquer um dos n homens considera aceitável qualquer uma das n mulheres, e vice-versa, embora possa preferir umas a outras. Nestas situações há a possibilidade de reformular o conceito de estabilidade, tendo Irving [14] introduzido três noções distintas:

- Estabilidade *fraca*: não existe um par $(h, m) \notin E$ tal que h prefere estritamente m à sua parceira em E e m prefere estritamente h ao seu parceiro em E.

- Estabilidade *forte*: não existe um par $(h, m) \notin E$ tal que pelo menos um dos dois prefere estritamente o outro ao seu par em E, e o outro também o prefere estritamente ou de igual modo.

- Estabilidade *super*: não existe um par $(h, m) \notin E$ tal que algum prefere o outro pelo menos tanto quanto prefere o seu par em E.

Uma instância de StableMarriageWithTies, com preferências não estritas (empates) mas listas completas, pode não admitir qualquer emparelhamento fortemente estável ou super estável [14]. O exemplo trivial é o caso em que cada mulher manifesta igual preferência por todos os homens, e cada homem manifesta igual preferência por todas as mulheres. Em [14], Irving apresenta um algoritmo de complexidade $O(n^2)$ que permite decidir se uma dada instância de StableMarriageWithTies admite algum emparelhamento fortemente estável ou super estável e, em caso afirmativo, determiná-lo. É também conhecido que qualquer instância desse problema admite sempre um emparelhamento com estabilidade fraca, bastando resolver os empates por especificação duma ordem arbitrária que os quebre, e aplicar o Algoritmo de Gale-Shapley. Observe-se que, qualquer que seja a instância particular de StableMarriageWithTies considerada, todos os emparelhamentos estáveis têm exactamente n pares, desde que exista algum emparelhamento estável (segundo o critério pretendido).

3.1.1 Listas de preferências incompletas e ordem não estrita

A situação é bastante diferente para StableMarriageWithTiesAndIncompleteLists. Por um lado, é fácil construir instâncias para as quais existem emparelhamentos fracamente estáveis que não têm o mesmo número de pares ou que diferem nos grupos de elementos que ficam emparelhados e sem par. Recorde-se que isto não se verificava no caso de StableMarriageWithIncompleteLists. Por outro lado, os trabalhos de Iwana et al. [15] e de Manlove et al. [19] mostram, que, devido a esse facto, alguns problemas passam a ser NP-completos (*NP-complete*) e NP-difíceis (*NP-hard*), o que, à partida, não augura nada de bom quanto à possibilidade de resolver computacionalmente instâncias genéricas desses problemas.

Entre esses problemas incluem-se o da determinação dum emparelhamento estável que envolva um número máximo (ou mínimo) de pares e o da verificação da estabilidade dum dado par (i.e., verificar se pode ocorrer em algum emparelhamento estável). Tal acontece mesmo num caso mais restritivo, em que a indiferença se traduz por empates num dos lados apenas, esses empates ocorrem no fim das listas de preferências, existindo quando muito um empate por lista, o qual tem comprimento dois (ou seja, envolve apenas um par de escolhas, pelas quais o interveniente manifestou igual preferência). A noção de estabilidade é a de *estabilidade fraca*.

4 Concurso de Colocação de Professores em Portugal

O Decreto-Lei nº 35/2003, de 27 de Fevereiro, republicado no Diário da Républica - 1ª Série A, nº 13, de 19 Janeiro de 2005, que no essencial regula actualmente o Concurso de Educadores de Infância e de Professores dos Ensinos Básico e Secundário, em Portugal, nos números 1 e 3 do artigo 12º, estabelece que

> *Art. 12º, nº1 - Os candidatos manifestam as suas preferências, por ordem decrescente de prioridade, por estabelecimentos de educação ou de ensino, por concelhos e por quadros de zona pedagógica.*

> *Art. 12$^{\underline{o}}$, n$^{\underline{o}}$3 - Quando os candidatos indicarem códigos de concelhos, considera-se que manifestam igual preferência por todos os estabelecimentos de educação ou de ensino de cada um desses concelhos, excepto pela escola de vinculação do candidato, que se considera excluída da preferência.*

e no número 2 do artigo 36$^{\underline{o}}$, referente ao concurso de afectação para professores de quadros de zona pedagógica (abreviadamente QZP), que

> *Art. 36$^{\underline{o}}$, n$^{\underline{o}}$2 - Quando a candidatura não esgote a totalidade dos estabelecimentos de educação ou ensino, considera-se que manifesta igual preferência por todos os restantes estabelecimentos.*

isto é, por todos os restantes estabelecimentos do QZP a que pertence o candidato.

Face ao exposto anteriormente, e ao facto deste concurso poder ser visto como uma instância de StableMarriageWithTiesAndIncompleteLists (ou NRMP com empates), surge a questão de saber se a actual legislação permite que uma solução do problema (isto é, uma colocação justa dos candidatos) possa ser obtida em tempo útil. Para ver que se trata duma instância de StableMarriageWithTiesAndIncompleteLists note-se que as escolas (ou vagas) correspondem aos hospitais (ou mulheres) e os opositores ao concurso correspondem aos internos (ou homens). A diferença essencial está na existência duma lista de graduação dos opositores (com prioridades), o que de certo modo, faz com que, todas as escolas (i.e., mulheres) tenham exactamente as mesmas preferências pelos candidatos (i.e., homens). Este facto pode ser determinante para a resolução do problema, porque permite concluir que esta instância de StableMarriageWithTiesAndIncompleteLists tem complexidade polinomial, como se mostra na Secção 5.

Sempre que adiante se mencionar a **lista de graduação**, está sempre subjacente também o respeito pelas prioridades com que os opositores concorrem, por vezes escrevendo-se explicitamente *lista de graduação (com prioridades)* ou **lista ordenada**. Um candidato que ocorra primeiro do que outro nessa lista, será considerado **mais graduado** do que esse.

4.1 Algoritmo desenvolvido pela ATX Software

O algoritmo para destacamento de docentes do quadro desenvolvido pela ATX Software [2, 8] para resolver o problema da colocação de professores no ano escolar de 2004/05, e que esta divulgou e disponibilizou ao público no seu *site*, foi o seguinte.

Algoritmo da ATX Software (para destacamento)

> Considerar inicialmente também livres as posições iniciais dos professores que pretendem mudar de posição.
> Colocar os professores pela ordem da lista de graduação, na melhor preferência ainda livre de cada professor (alguns professores podem ficar por colocar).
> Se os professores que estavam inicialmente colocados ficaram todos colocados, o processo termina.
> Senão,
>> Os professores que estavam inicialmente colocados e ficaram por colocar são colocados definitivamente nas suas posições iniciais, que deixam de estar livres. Estes professores já não entram nas próximas iterações.
>> Repete-se a colocação com menos estes lugares livres.

Em [2], a empresa demonstra a correcção do algoritmo e refere que a sua complexidade computacional é $O(C^2V)$, em que C é o número de candidatos e V o número de vagas. Refere ainda que *para o caso de docentes contratados, as regras de colocação não são exactamete as mesmas, pelo que foi desenvolvido um algoritmo ligeiramente diferente* [2].

No desenvolvimento do algoritmo, foi pressuposto que cada candidato define uma lista de preferências (por vagas) juntamente com uma ordem total. Assim, tendo em conta o exposto na Secção 3, e o facto de existir uma lista de graduação totalmente ordenada, tratar-se-ia então da resolução duma instância de StableMarriageWithIncompleteLists. Nesse pressuposto, o algoritmo apresentado é de facto correcto, como a ATX Software demonstrou. Pode-se mesmo concluir que é possível baixar a complexidade do algoritmo de resolução para $O(CV)$ se se adoptar antes o Algoritmo de Gale-Shapley (orientado por internos).

Basta para tal considerar, grosso modo, que cada escola tem preferência estrita (e idêntica) por todos os seus docentes que são opositores ao concurso de destacamento e a seguir por todos os restantes, segundo a lista ordenada. E, acrescentar à lista de preferências de cada docente, que está a concorrer para destacamento, a sua escola (vaga) actual, como última preferência.

Realizando-se as colocações por afectação e destacamento (com excepção dos destacamentos por ausência de serviço educativo) simultaneamente, como estabelece o número 8 do artigo 30º do Decreto-Lei nº 35/2003, de 27 de Fevereiro, apenas os opositores à afectação podem ficar sem colocação. Tal fica garantido pela inclusão das posições actuais dos opositores a destacamento nas respectivas listas de preferências, como se acabou de descrever. Quando um desses docentes se propuser à sua vaga (por já ter esgotado as escolhas da sua preferência), esta aceita-o incondicionalmente.

4.2 Unicidade da solução

A noção de emparelhamento estável (estabilidade fraca) pode ser identificada com o estabelecido no número 2 do artigo 18º e no número 1 do artigo 24º do Decreto-Lei nº 35/2003, de 27 de Fevereiro, nomeadamente:

> *Art. 18º, nº 2 - O preenchimento das vagas e dos horários respeita as preferências identificadas no presente diploma e a lista definitiva de ordenação e manifesta-se através de listas de colocações, as quais dão origem igualmente a listas graduadas de candidatos não colocados (...)*
>
> *Art. 24º, nº 1 - Os concursos realizam-se com recuperação automática de vagas, de modo que cada candidato não seja ultrapassado em qualquer das suas preferências por outro com menor graduação na mesma prioridade.*

No pressuposto de que o conjunto de preferências de cada docente está totalmente ordenado, bem como o conjunto de docentes, pode-se demonstrar que existe uma única lista de colocações possível, isto é um único emparelhamento estável, **se se assumir que as escolas não vão tentar reter os seus docentes, se estes conseguirem destacamento**[4].

De facto, é conhecido que qualquer instância de StableMarriage em que todas as mulheres têm exactamente a mesma lista de preferências admite um único emparelhamento estável [13]. O homem i nessa lista de preferências é emparelhado com a sua favorita dentre

[4] Se p_1 e p_2 estivessem colocados em v_1 e v_2 actualmente e p_2 pretendesse v_1 e p_1 pretendesse v_2, a solução $\{(p_1, v_1), (p_2, v_2)\}$ seria estável, mas as escolas estavam a reter os seus docentes, não os deixando ficar nas vagas que pretendiam. Esta colocação não será considerada admissível. Gostaria de agradecer a João Pascoal Faria a detecção desta imprecisão na noção de estabilidade que introduzi na versão original deste relatório.

as que ficam livres depois de se ter já emparelhado os homens $1, \ldots, i-1$. O resultado é válido ainda para as instâncias de StableMarriageWithIncompleteLists em que todas as mulheres têm a mesma lista de preferências, embora neste caso o homem i possa agora ficar sem par se o emparelhamento dos anteriores $1, \ldots, i-1$, esgotar a lista de preferências de i. Esta lista única de preferências de todas as mulheres corresponde no problema de colocação de professores à lista de graduação (com prioridades).

No caso dos destacamentos, as vagas de opositores ao concurso só são efectivamente libertadas se estes conseguirem o destacamento. Como acima se descreveu, cada uma destas vagas condicionais, dá, de certo modo, prioridade ao seu actual detentor, face aos restantes candidatos. Se este não conseguir destacamento, acaba por se propor à sua vaga, e esta aceita-o necessariamente, rejeitando qualquer candidato que tenha sido nela colocado provisoriamente. Supõe-se aqui que se estaria a usar o Algoritmo de Gale-Shapley (orientado por internos) para determinar a lista de colocações, cada vaga correspondendo a um hospital.

Mesmo com esta preferência estrita de algumas vagas por alguns candidatos, é possível continuar a mostrar a existência duma lista de colocações única, quando se assume que as escolas não vão tentar reter os seus docentes, se estes conseguirem destacamento. Por análise, por exemplo, do Algoritmo da ATX Software (para destacamento), pode-se concluir facilmente que à medida que os professores que não conseguem destacamento vão sendo retirados, bem como as suas vagas, as listas de preferências das vagas vão-se aproximando da lista de ordenada de candidatos. Assim, se conclui que existirá apenas uma lista de colocações possível, à semelhança do que acontece no StableMarriageWithIncompleteLists, quando todas as mulheres têm a mesma lista de preferências.

Para o concurso relativo ao ano escolar 2004/05 o número 4 do artigo 41º estabelecia:

> *Art. 41º, nº4 - Para efeitos de destacamento ao abrigo da preferência conjugal, os candidatos podem concorrer aos estabelecimentos de educação ou de ensino do concelho onde se situa a residência familiar ou o local onde o cônjuge exerça ou venha a exercer a sua actividade profissional no ano escolar a que o concurso respeita, não podendo o número de estabelecimentos indicados exceder 50 (...)*

pelo que, talvez se possa concluir que, estes docentes não podem indicar apenas o concelho nele mencionado, e desta forma concorrer com igual preferência a todas as escolas do mesmo. Nada de análogo era dito sobre os docentes opositores ao concurso de destacamento por condições específicas[5].

O que se entende por "manifestar igual preferência"? O Decreto-Lei 35/2003, de 27 de Fevereiro, não pressupõe que as listas de preferências dos candidatos estejam totalmente ordenadas.

Não se pode afirmar que estão totalmente ordenadas as listas de preferências dos opositores voluntários a destacamento por ausência de serviço educativo, dos docentes de QZP em afectação, dos opositores ao concurso de contratação, e dos opositores ao concurso de provimento. No entanto, o aviso de abertura do concurso (Aviso nº 2598-B/2004 (2ª série), publicado no Diário da República nº49/2004, 27 de Fevereiro), no capítulo III, estabelece, para afectação de docentes de QZP, o seguinte:

[5] Mas, de facto, por adenda ao referido Decreto-Lei e detalhe no aviso de abertura do Concurso para o ano escolar de 2005/06, são acrescentadas algumas restrições para manifestação de preferências, na linha das do destacamento ao abrigo da preferência conjugal.

1.4.3 - Os docentes que não manifestam preferências são afectos obrigatoriamente a uma das escolas do âmbito geográfico do respectivo quadro de zona pedagógica.

e no capítulo VI, estabelece

f) Campo 7 – Manifestação de preferências para provimento
f.3) A manifestação de preferências por lugares de quadro de escola ou de zona pedagógica obedece ao disposto no artigo 12º do Decreto-Lei nº 35/2003, podendo os candidatos individualizar por ordem decrescente de prioridade, até 50 códigos de estabelecimentos de educação ou de ensino, 25 códigos de concelhos, e 23 códigos de quadro de zona pedagógica correspondentes à sua totalidade, podendo alternar ou conjugar as preferências;
(. . .)
f.5) Quando o candidato identificar códigos de concelhos, considera-se que manifesta igual preferência por todos os estabelecimentos de educação ou de ensino de cada um, percorrendo-se os códigos das escolas respectivas, por ordem crescente até obtenção de colocação. No entanto, logo que outro candidato liberte vaga em alguma das escolas a que tiver sido conferida melhor preferência, é esta a colocação definitiva;
(. . .)
g) Campo 8 – Manifestação de preferências para contratação
(. . .)
g.4) À identificação de códigos de concelhos e de quadros de zona pedagógica corresponde, de acordo com os nºs 3 e 4 do artigo 12º, a manifestação de igual preferência por todos os estabelecimentos de educação ou de ensino do respectivo âmbito geográfico;

Esta ordenação estrita das preferências que fica determinada pela alínea *f.5)* deste Aviso[6], pode ser bastante desfavorável para os candidatos (individual e colectivamente). O mesmo se aplica às colocações por afectação, destacamento por condições específicas, e destacamento ao abrigo da preferência conjugal (e outro), as quais decorrem em simultâneo de acordo com o Decreto-Lei nº 35/2003, ainda que com prioridade das primeiras face às segundas e destas face às terceiras[7]. Mesmo que as listas de preferências dos opositores a destacamento para os dois tipos referidos estejam totalmente ordenadas, no algoritmo usado para a afectação, deveria ter que se considerar soluções alternativas, pois a afectação acaba por poder ter interferência nos destacamentos (cf., Exemplo 2 da próxima secção). Mas não foi feito isso. Por outro lado, no pressuposto de que a lista ordenada de professores reflecte dalgum modo a sua graduação e competência, esta afectação de professores a escolas, dando "preferência" às escolas que têm um código mais baixo, faz com que estas fiquem com os melhores professores dentre os que a elas se candidatam[8]. E porquê umas escolas e não outras?

4.3 Efeitos perversos da ordenação estrita das listas de preferências

Os exemplos seguintes mostram que se podem facilmente construir instâncias hipotéticas do problema que não têm solução única. Mostram ainda que diferentes ordenações das vagas

[6] No Aviso nº 1413-B/2005 (2ª série), 11 de Fevereiro, relativo à abertura do concurso para 2005/06, este artigo deixou de constar.

[7] No concurso relativo ao ano escolar 2004/05, o destacamento por condições específicas tinha prioridade relativamente à afectação.

[8] Agradeço a Michel Ferreira, DCC-FC & LIACC, UP, este comentário.

ou lugares a concurso, que se efectuem para se assumir que as listas de preferências estão totalmente ordenadas, podem conduzir a listas de colocações diferentes, nem todas colocando exactamente os mesmos candidatos até ao mesmo número de ordem da lista ordenada, por exemplo.

Afectação e destacamento em simultâneo. O Exemplo 2 mostra que se, como estabelece o número 8 do artigo 30º do Decreto-Lei nº 35/2003, as colocações por afectação e destacamentos decorrerem em simultâneo, se podem recuperar automaticamente vagas. Ilustra também que, se ordenar estritamente as listas de preferências dos candidatos, se podem perder algumas soluções e obter listas de colocações piores. O Exemplo 3 mostra que, se as referidas colocações forem efectuadas sequencialmente (como parece indicar o artigo 4º do capítulo XIX do Aviso nº 1413-B/2005 (2ª série), de 11 de Fevereiro, relativo à abertura do Concurso), pode haver casos em que alguns candidatos não vão conseguir posições que preferiam e outros não vão conseguir destacamento.

Exemplo 2 *Três candidatos p_1, p_2 e p_3 apresentam as seguintes listas de preferências*

p_1: *QZP (i.e., v_3, v_1 e v_2 com igual preferência)*
p_2: v_3, v_2 *(colocado em v_1)*
p_3: v_1 *(colocado em v_3)*

estando p_1 a concorrer para afectação em QZP, p_2 para destacamento por condições específicas, e p_3 a destacamento ao abrigo da preferência conjugal. As únicas posições que restam no QZP são v_3, v_2 e v_1, e não se situam no mesmo concelho. Só v_2 está livre, pois p_2 está colocado em v_1 e p_3 em v_3. Neste caso, o emparelhamento

$$\{(p_1, v_2), (p_2, v_3), (p_3, v_1)\}$$

é estável e está de acordo com número 8 do artigo 30º do Decreto-Lei nº 35/2003.

Se as listas de preferências fossem

p_1: v_1, *QZP (isto é, v_3 e v_2 com igual preferência)*
p_2: v_3, v_2 *(colocado em v_1)*
p_3: v_1 *(colocado em v_3)*

então $\{(p_1, v_2), (p_2, v_3), (p_3, v_1)\}$ já não seria um emparelhamento estável, porque p_1 tem direito ao lugar v_1, se este for libertado. Se se assumir que as escolas não tentam reter os seus docentes, o único emparelhamento estável seria $\{(p_1, v_1), (p_2, v_2), (p_3, v_3)\}$ e p_3 não conseguiria destacamento.

Se, para se considerar que as listas de preferências estão totalmente ordenadas, se identificasse os dois casos com

p_1: v_1, v_3, v_2
p_2: v_3, v_2 *(colocado em v_1)*
p_3: v_1 *(colocado em v_3)*

então, no mesmo pressuposto, a única solução estável seria $\{(p_1, v_1), (p_2, v_2), (p_3, v_3)\}$, como anteriormente, e p_3 não conseguiria destacamento.

Exemplo 3 *Três candidatos p_1, p_2 e p_3 apresentam as seguintes listas de preferências*

p_1: *QZP*
p_2: v_2 *(colocado em v_1)*
p_3: v_1 *(colocado em v_3)*

estando p_1 a concorrer para afectação em QZP, p_2 para destacamento por condições específicas, e p_3 a destacamento ao abrigo da preferência conjugal. As posições v_3, v_2 e v_1 situam-se em três concelhos distintos desse QZP e só v_2 está livre. Se se começasse por efectuar as colocações por afectação e só depois os destacamentos, p_1 ocuparia v_2 e nem p_2 nem p_3 conseguiriam destacamento. Se a afectação e destacamento decorrerem em simultâneo, então $\{(p_1, v_3), (p_2, v_2), (p_3, v_1)\}$ é uma solução (e é óptima, segundo os candidatos).

Destacamento de docentes por ausência de serviço. Nos termos do número 3 do artigo 32º do Decreto-Lei, *para efeitos de destacamento voluntário, podem os docentes manifestar as suas preferências de acordo com o disposto no artigo 12º*. O Exemplo 4 mostra que se as listas de preferências forem ordenadas arbitrariamente, alguns docentes podem ficar por destacar.

Exemplo 4 *Quatro docentes p_1, p_2, p_3 e p_4 em destacamento voluntário por ausência de serviço educativo, apresentam as seguintes listas de preferências*

p_1: *concelho c*
p_2: *concelho c*
p_3: v_1
p_4: v_2

sendo v_1, v_2, v_3, v_4 as únicas quatro vagas nesse concelho (todas em escolas diferentes), p_1 o primeiro professor na lista ordenada e p_4 o último. Note-se que o número 3 do artigo 32º do referido Decreto-Lei, não impõe qualquer número mínimo de escolhas a estes candidatos nem outras restrições para além das estabelecidas pelo artigo 12º. Assim, hipoteticamente, o exemplo anterior parece fazer sentido, decorrendo do Decreto-Lei que p_1 e p_2 estão a concorrer com igual preferência a todas as vagas de c.

Se se considerar que as vagas são arbitrariamente ordenadas, o que corresponde ao pressuposto de que as listas de preferências dos docentes estão totalmente ordenadas, pode-se obter emparelhamentos diferentes. Se as preferências de p_1 e p_2 forem entendidas como v_1, v_2, v_3 e v_4, por esta ordem, a lista de colocações ficaria

$$\{(p_1, v_1), (p_2, v_2), (p_3, \textit{não destacado}), (p_4, \textit{não destacado})\}$$

e se forem entendidas, por exemplo, como v_3, v_4, v_2 e v_1, por esta ordem, a lista de colocações passaria a ser $\{(p_1, v_3), (p_2, v_4), (p_3, v_1), (p_4, v_2)\}$. Ambas as listas contemplam o que estabelece o número 2 do Art. 18º, mas são diferentes. Coloca-se então a questão de saber qual das duas é a melhor.

Afectação de docentes de QZP. O Exemplo 5 mostra novamente que se se ordenar as posições pelas quais os docentes manifestaram igual preferência, de forma que as suas listas de preferências passem a estar totalmente ordenadas, se podem obter listas de colocações piores, segundo os docentes. É de referir que, como a actual legislação obriga os professores de QZP a aceitarem qualquer vaga que lhes seja atribuída dentro do seu QZP, não é possível construir instâncias em que um professor mais graduado (i.e., que surge primeiramente na lista ordenada) não ficasse colocado no seu QZP e outro menos graduado ficasse. Recorde-se que todos os lugares (vagas) do QZP omitidos são considerados aceitáveis pelo candidato, e como se, por todos, manifestasse igual preferência.

Exemplo 5 *Suponha-se que quatro docentes p_1, p_2, p_3 e p_4 do mesmo grupo de docência e QZP se apresentam a afectação com as seguintes listas de preferências:*

p_1: *concelho c*
p_2: *concelho c*
p_3: v_1, *concelho c*
p_4: v_2, *concelho c*

sendo v_1, v_2, v_3, v_4 as únicas quatro vagas nesse concelho (todas em escolas diferentes), não se indicando as restantes vagas do QZP a que todos pertencem. Recorde-se que, pelo número 2 do artigo 36º, se considera que a não manifestação de preferência explícita por essas restantes vagas equivale à manifestação de igual preferência por todas elas.

Como no exemplo anterior, p_1 é o primeiro professor na lista ordenada e p_4 o último. O artigo 12º não impede que um docente, como p_3 e p_4, possa indicar preferência estrita por uma escola dum dado concelho e em seguida por esse concelho (entendendo-se, talvez, que está agora a manifestar preferência idêntica por todas as restantes escolas desse concelho).

Mais uma vez, se se ordenar arbitrariamente as vagas, o que corresponde ao pressuposto de que as listas de preferências dos docentes estão totalmente ordenadas, pode-se obter emparelhamentos diferentes. Caso as preferências de p_1 e p_2 sejam entendidas como v_1, v_2, v_3 e v_4, por esta ordem, a lista de colocações ficaria

$$\{(p_1, v_1), (p_2, v_2), (p_3, v_3), (p_4, v_4)\}$$

mas se forem entendidas, por exemplo, como v_3, v_4, v_2 e v_1, por esta ordem, a lista de colocações passaria a ser $\{(p_1, v_3), (p_2, v_4), (p_3, v_1), (p_4, v_2)\}$. Ambas as listas contemplam o estabelecido no número 2 do artigo 18º, mas enquanto que na segunda lista de colocação todos os candidatos ficam colocados na sua primeira escolha, na primeira lista, não é assim.

Concurso de contratação. A situação seria teoricamente grave se se adoptasse o mesmo pressuposto de que as listas dos candidatos estão totalmente ordenadas, também para efectuar as colocações nos concursos de provimento e de contratação. Neste caso, alguns candidatos poderiam ficar sem colocação quando, se fosse procurada a ou uma das melhores soluções, podiam ficar colocados. O Exemplo 6, mais uma vez hipotético, ilustra o que poderia acontecer.

Exemplo 6 *Seis professores $p_1, p_2, p_3, p_4, p_5, p_6$, todos do mesmo grupo de docência e todos opositores ao concurso externo, têm as seguintes listas de preferências:*

p_1: *concelho c*
p_2: *concelho c*
p_3: v_1
p_4: v_2
p_5: v_3
p_6: v_4

sendo v_1, v_2, v_3, v_4 as únicas quatro vagas nesse concelho (pertencendo a escolas diferentes), e está-se a proceder à colocação por contratação.

Se as preferências de p_1 e p_2 forem entendidas como v_1, v_2, v_3 e v_4, por esta ordem (para satisfação do pressuposto de que as listas de preferências estão totalmente ordenadas), a lista de colocações ficaria

$$\{(p_1, v_1), (p_2, v_2), (p_3, \textit{não contratado}), (p_4, \textit{não contratado}), (p_5, v_3), (p_6, v_4)\}$$

e se forem entendidas, por exemplo, como v_3, v_4, v_2 e v_1, por esta ordem, a lista de colocações passaria a ser $\{(p_1, v_3), (p_2, v_4), (p_3, v_1), (p_4, v_2), (p_5, \textit{não contratado}), (p_6, \textit{não contratado})\}$. Mais uma vez, ambas as listas contemplam o que estabelece o número 2 do artigo 18º, mas são diferentes. Como o tempo de serviço é relevante para a graduação dos candidatos, não só, a primeira solução é penalizadora para p_3 e p_4 por não terem conseguido colocação, e eventualmente, acabarem desempregados, mas também porque, no concurso seguinte, poderão acabar por ser ultrapassados por p_5 e p_6.

4.4 Lista de colocações óptima segundo os opositores ao concurso

Designe-se por $o_p(v)$, o número de ordem de v na lista de preferências do opositor p. Ou seja, $o_p(v)$ indica se v é 1ª, 2ª, 3ª, ... preferência de p. Todas aquelas vagas (lugares) pelas quais p manifesta igual preferência terão o mesmo número de ordem. Caso p não fique colocado, pode-se definir como última preferência a "não colocação" e atribuir-se-lhe o número de ordem imediatamente a seguir ao da última preferência da lista de preferências reais. No caso de destacamentos essa última preferência fictícia seria a actual posição de p.

Se se analisar a sequência de $o_p(v)$'s seguindo a ordem da lista ordenada pode-se perceber o que é uma *lista de colocações óptima* do ponto de vista dos opositores ao concurso. Em termos formais, é uma lista de colocações para a qual a referida sequência de $o_p(v)$'s é lexicograficamente[9] mínima, dentre as que correspondem a soluções estáveis. Podem existir várias listas óptimas, mas todas definem exactamente a mesma sequência de $o_p(v)$'s, pois a ordem lexicográfica é uma ordem total.

Exemplo 7 *Voltando novamente a considerar a situação descrita no Exemplo 6,*

p_1: *concelho c*
p_2: *concelho c*
p_3: v_1
p_4: v_2
p_5: v_3
p_6: v_4

[9] $(x_1, \ldots, x_n) <_{LEX} (y_1, \ldots, y_n)$ sse existir i com $1 \leq i \leq n$ tal que $x_i < y_i$ e $x_k = y_k$ para todo $k < i$.

a lista $\{(p_1, v_1), (p_2, v_2), (p_3, \textit{não contratado}), (p_4, \textit{não contratado}), (p_5, v_3), (p_6, v_4)\}$ *definiria a sequência (1,1,2,2,1,1), enquanto que*

$$\{(p_1, v_3), (p_2, v_4), (p_3, v_1), (p_4, v_2), (p_5, \textit{não contratado}), (p_6, \textit{não contratado})\}$$

definiria (1,1,1,1,2,2), que é lexicograficamente inferior. Pode-se concluir que é óptima uma vez que só estão disponíveis quatro vagas, e esta sequência tem prefixo 1,1,1,1. A lista

$$\{(p_1, v_4), (p_2, v_3), (p_3, v_1), (p_4, v_2), (p_5, \textit{não contratado}), (p_6, \textit{não contratado})\}$$

definiria exactamente a mesma sequência (1,1,1,1,2,2), constituindo uma solução óptima alternativa. Não é difícil concluir que existiriam apenas estas duas listas óptimas de colocações.

Na eventualidade dalguma escola não conseguir preencher as suas vagas na fase de contratação, estas podem sempre transitar para a oferta de escola. Deste modo, a opção por listas de colocações óptimas segundo os opositores ao concurso não penaliza as escolas, no seu conjunto.

5 Algoritmos Polinomiais para Problemas com Indiferença

Nesta secção mostra-se a existência de algoritmos polinomiais para determinar listas de colocações óptimas, segundo os opositores ao concurso de colocação de professores, satisfazendo naturalmente as restrições impostas pela legislação actual.

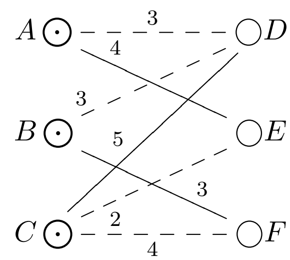
A prova deste resultado será efectuada por redução do problema a uma sequência de problemas de determinação dum emparelhamento perfeito com peso máximo num grafo bipartido. Cada um destes problemas pode ser resolvido em tempo polinomial [6, 18, 21, 25], por exemplo, pelo Algoritmo de Kuhn-Munkres, também conhecido por Método Húngaro [18, 21], que é de $O(n^4)$, sendo n o número de vértices do grafo. Pode-se baixar esta complexidade, por exemplo, para $O(n^3)$, se se tirar partido de propriedades estruturais das soluções [6, 25].

Em termos matemáticos, o problema da determinação dum emparelhamento com peso máximo num grafo bipartido $G = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{A}, c)$, com $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$, pode ser formulado como

$$\left|\begin{array}{l} \text{maximizar} \displaystyle\sum_{e \in \mathcal{A}} c_e x_e \\ \text{sujeito a} \\ \quad \sum_{e \in \delta(i)} x_e \leq 1, \text{ para todo } i \in \mathcal{V}_1 \cup \mathcal{V}_2. \\ \quad x_e \in \{0, 1\}, \text{ para todo } e \in \mathcal{A} \end{array}\right.$$

onde $\delta(i)$ denota o conjunto de ramos incidentes no vértice i, ou seja, tais que i é seu extremo. A variável de decisão x_e toma valor 1 se e só se o ramo e pertencer ao emparelhamento determinado, com $e \in \mathcal{A}$. Como habitualmente, supõe-se que $\mathcal{V}_1 \cup \mathcal{V}_2$ é o conjunto de vértices do grafo, cada ramo $e \in \mathcal{A}$ tem um extremo em $\mathcal{V}_1$ e outro em $\mathcal{V}_2$, e $c : \mathcal{A} \mapsto \mathbb{R}_0^+$ define o peso associado a cada ramo. Na instância que se irá considerar, $|\mathcal{V}_1| = |\mathcal{V}_2|$ e o emparelhamento será perfeito, isto é terá um ramo incidente em cada vértice do grafo.

Exemplo 8 *Para ilustrar o conceito, apresenta-se o grafo seguinte*



no qual $\{(A, E), (B, F), (C, D)\}$ *constituiu um emparelhamento perfeito com peso máximo (neste caso,* $4+3+5 = 12$*). Os ramos do grafo que não pertencem ao emparelhamento estão indicados a tracejado.*

5.1 Relação entre problemas de colocação e afectação clássica com pesos

Sejam $p_1, \ldots, p_n$ os candidatos e $v_1, \ldots, v_k$ as posições a concurso. Estas posições incluem as dos candidatos que detêm já uma posição, a qual libertarão se ficarem colocados noutra da sua preferência. Considere-se os conjuntos de vértices $\mathcal{V}_1 = \{p_1, \ldots, p_n\} \cup \{p'_1, \ldots, p'_k\}$ e $\mathcal{V}_2 = \{v_1, \ldots, v_k\} \cup \{v'_1, \ldots, v'_n\}$, em que $p'_1, \ldots, p'_k$ denotam candidatos fictícios e $v'_1, \ldots, v'_n$ posições fictícias. A interpretação a ser dada a estes candidatos e posições fictícias deduz-se da seguinte definição para o conjunto $\mathcal{A}$ de ramos do grafo:

- $(p_i, v_j) \in \mathcal{A}$ se p_i estiver a concorrer à posição v_j, com $1 \leq i \leq n$ e $1 \leq j \leq k$. Se (p_i, v_j) pertencer ao emparelhamento, o candidato p_i fica colocado na posição v_j.

- $(p_i, v_j) \in \mathcal{A}$ se p_i detem actualmente a posição v_j, à qual terá direito se não conseguir outra colocação no concurso, com $1 \leq i \leq n$. Se (p_i, v_j) pertencer ao emparelhamento, o candidato p_i não alterou a sua posição.

- $(p_i, v'_i) \in \mathcal{A}$ se p_i não detem actualmente posição, para $1 \leq i \leq n$. Se (p_i, v'_i) pertencer ao emparelhamento, p_i não ficou colocado.

- $(p'_j, v_j) \in \mathcal{A}$ com $1 \leq j \leq k$. Se (p'_j, v_j) pertencer ao emparelhamento, então a posição v_j não fica atribuída.

- $(p'_j, v'_i) \in \mathcal{A}$ com $1 \leq j \leq k$ e $1 \leq i \leq n$. Se (p'_j, v'_i) pertencer ao emparelhamento, a posição v_j não ficou livre (e o professor p_i ficou colocado). Note-se que, pela definição de $\mathcal{A}$, se (p'_j, v_j) não estiver no emparelhamento então v_j está atribuída a algum dos professores reais. Por outro lado, por definição de emparelhamento, não se pode ter (p'_j, v_j) e (p'_j, v'_i) simultaneamente num emparelhamento.

Observe-se que o grafo admite pelo menos um emparelhamento perfeito, por exemplo, o emparelhamento constituído pelos pares seguintes:

- (p_i, v_j), (p'_j, v'_i) se v_j for a posição actual de p_i, para $1 \leq i \leq n$;

- (p_i, v'_i) se p_i não tiver actualmente posição, para $1 \leq i \leq n$;

- (p'_j, v_j) se a posição v_j está livre actualmente, $1 \leq j \leq k$.

o qual, poderá não ser estável. Para que a definição do grafo $G = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{A}, c)$ fique completa, é necessário indicar os pesos associados aos ramos, ou seja, definir a função c. Os pesos terão que ser definidos de forma a garantir que os emparelhamentos que têm peso máximo sejam emparelhamentos estáveis e óptimos segundo os candidatos.

Para cada p_i, seja $\Gamma_{i,1}, \Gamma_{i,2}, \ldots, \Gamma_{i,\gamma_i}$ a sua lista de preferências, ordenada por ordem estritamente decrescente, $\Gamma_{i,t}$ denotando um conjunto de posições pelas quais p_i tem igual preferência, para todo $1 \leq t \leq \gamma_i$.

Junte-se ao final dessa sequência de preferências, a posição actual do candidato p_i se este estiver colocado e, caso contrário, v'_i. Ou seja, a lista de preferências seria

$$p_i : \begin{cases} \Gamma_{i,1}, \Gamma_{i,2}, \ldots, \Gamma_{i,\gamma_i}, \{v_j\}, & \text{se } v_j \text{ é a posição actual de } p_i \\ \\ \Gamma_{i,1}, \Gamma_{i,2}, \ldots, \Gamma_{i,\gamma_i}, \{v'_i\}, & \text{se } p_i \text{ não tem posição actualmente} \end{cases}$$

passando a ser referida essa última posição por Γ_{i,γ_i+1}. No final do concurso, p_i fica colocado nalguma das posições que constam desta lista.

Observe-se que se $v \in \Gamma_{i,t}$, então t é o número de ordem da posição v na lista de p_i, isto é $o_{p_i}(v) = t$, para $1 \leq t \leq \gamma_i + 1$. Será atribuído o mesmo peso $c(p_i, \Gamma_{i,t})$ às posições que pertencem ao mesmo conjunto $\Gamma_{i,t}$. Este peso representa o valor da colocação de p_i nalguma das posições de $\Gamma_{i,t}$.

5.1.1 Colocações por contratação como um problema de afectação com pesos

No caso do concurso de contratação, em que *nenhum candidato está afecto a qualquer posição anterior*, isto é, $\Gamma_{i,\gamma_i+1} = \{v'_i\}$ para todo i, pode-se tomar

$$c(p_i, \Gamma_{i,t}) = 1 + (\gamma_i + 1 - t) r_i$$

ou seja,

$$c(p_i, v) = 1 + (\gamma_i + 1 - t) r_i, \text{ para todo } v \in \Gamma_{i,t} \text{ e todo } t \text{ com } 1 \leq t \leq \gamma_i + 1.$$

Portanto, os valores $c(p_i, \Gamma_{i,t})$ são termos consecutivos duma progressão aritmética com termo inicial 1 e razão r_i. A razão pode ser definida por

$$r_i = \begin{cases} 1 & \text{se } i = n \\ \displaystyle\sum_{i'=i+1}^{n} c(p_{i'}, \Gamma_{i',1}) - (n - i) + 1 & \text{se } 1 \leq i < n \end{cases}$$

de forma a garantir a estabilidade dos emparelhamentos de peso máximo e a sua optimalidade do ponto de vista dos candidatos.

Note-se que $r_{i-1} = \sum_{i'=i}^{n} c(p_{i'}, \Gamma_{i',1}) - (n - (i-1)) + 1 = r_i + c(p_i, \Gamma_{i,1}) - 1 = r_i(\gamma_i + 1)$ ou seja,

$$r_{i-1} = r_i(\gamma_i + 1), \text{ para todo } i \leq n$$

o que implica um crescimento exponencial do valor da razão r_i e, consequentemente, dos pesos assim definidos.

Para os candidatos fictícios, define-se $c(p'_j, v_j) = 0$, para $1 \leq j \leq k$ e $c(p'_j, v'_i) = 0$, para $1 \leq j \leq k$ e $1 \leq i \leq n$. O peso destes ramos é zero porque, tal como os nós fictícios, só são introduzidos para conseguir restringir a procura a emparelhamentos perfeitos.

Exemplo 9 *Para a situação tratada nos Exemplos 6 e 7, para concurso de contratação com seis candidatos*

p_1*: concelho c*
p_2*: concelho c*
p_3*:* v_1
p_4*:* v_2
p_5*:* v_3
p_6*:* v_4

em que v_1, v_2, v_3, v_4 *são as únicas quatro posições disponíveis no concelho c, ter-se-ia, segundo o modelo introduzido*

$$\begin{array}{ll}
p_1 : c(p_1, \{v_1, v_2, v_3, v_4\}) = 33,\ c(p_1, \{v_1'\}) = 1 & r_1 = 32 \\
p_2 : c(p_2, \{v_1, v_2, v_3, v_4\}) = 17,\ c(p_2, \{v_2'\}) = 1 & r_2 = (9 + 5 + 3 + 2) - 4 + 1 = 16 \\
p_3 : c(p_3, \{v_1\}) = 9,\ c(p_3, \{v_3'\}) = 1 & r_3 = (5 + 3 + 2) - 3 + 1 = 8 \\
p_4 : c(p_4, \{v_2\}) = 5,\ c(p_4, \{v_4'\}) = 1 & r_4 = 3 + 2 - 2 + 1 = 4 \\
p_5 : c(p_5, \{v_3\}) = 3,\ c(p_5, \{v_5'\}) = 1 & r_5 = 2 - 1 + 1 = 2 \\
p_6 : c(p_6, \{v_4\}) = 2,\ c(p_6, \{v_6'\}) = 1 & r_6 = 1
\end{array}$$

existindo duas listas de colocações óptimas, ambas com valor $33 + 17 + 9 + 5 + 1 + 1 = 66$*:*

$$\{(p_1, v_3), (p_2, v_4), (p_3, v_1), (p_4, v_2), (p_5, v_5'), (p_6, v_6')\}$$
$$\{(p_1, v_4), (p_2, v_3), (p_3, v_1), (p_4, v_2), (p_5, v_5'), (p_6, v_6')\}.$$

Tendo em conta o número de candidatos, é de prever instâncias reais com dimensões consideráveis. A utilização directa deste modelo levanta, por isso, algumas dificuldades se os pesos forem calculados explicitamente. O número 2 do artigo 12º do Decreto-Lei nº35/2003, de 27 Fevereiro (republicação), estipula:

> *Art. 12º, nº2 - Na manifestação das suas preferências os candidatos devem indicar os códigos referidos nas alíneas seguintes, podendo quer alternar as preferências dessas alíneas, quer conjugar as preferências contidas em cada uma delas:*
> *a) Códigos de estabelecimentos de educação ou de ensino, no máximo de 75;*
> *b) Códigos de concelhos no máximo de 50;*
> *c) Códigos de zona pedagógica, no máximo dos quadros existentes.*

existindo actualmente 23 zonas pedagógicas, cada uma correspondendo a um agrupamento de escolas de um ou vários concelhos. De momento, o interesse do modelo proposto pode ser essencialmente teórico. Vai permitir demonstrar que o problema da determinação de listas de colocações óptimas pode ser resolvido em tempo polinomial *na fase de contratação*. Depois de adaptado, permitirá mostrar o mesmo para o problema geral, que contempla a existência de candidatos a *destacamento*.

Desde que se prove que o modelo proposto está correcto, pode-se concluir que, na fase de contratação, o problema se pode resolver em $O((n+k)^4\mathcal{K})$, por aplicação do Algoritmo de Kuhn-Munkres, na sua versão mais simples, ou em $O((n+k)^3\mathcal{K})$ com uma implementação melhor, sendo n o número de candidatos e k o de posições. Relativamente ao Algoritmo de Kuhn-Munkres, esta complexidade apresenta assim um agravamento dum factor $\mathcal{K}$, que depende da dimensão do problema, e que mais adiante se mostra ser linear em n. Tal agravamento resulta de não se poder assumir sempre que as operações aritméticas que o algoritmo envolve se realizam em tempo constante[10]. Os pesos introduzidos crescem exponencialmente com a dimensão do problema, a qual, para instâncias reais, não é negligenciável. Na Secção 5.3 analisa-se a função de peso proposta e mostra-se como a tratar sem comprometer a complexidade polinomial do algoritmo. Essa demonstração é suportada por aspectos técnicos, que, por questões da estrutura lógica do relatório, será mais oportuno abordar mais adiante.

Adaptando o modelo agora introduzido, mostra-se, na Secção 5.2, que o problema geral, em que há candidatos que já detêm uma posição inicial, pode ser resolvido por algoritmos polinomiais.

A seguir justifica-se a correcção do modelo que se acabou de propor.

Pesos garantem solução lexicograficamente mínima. Na fase de contratação, qualquer candidato p_i pode ficar "não colocado", isto é, pode ficar na posição v_i'. Esta propriedade garante que nessa fase, qualquer solução lexicograficamente mínima é estável. De facto, se uma dada *solução*[11] *lexicograficamente mínima* não fosse estável, então existia um par (p_i, v_j) tal que p_i preferia v_j à posição em que ficou colocado (segundo essa solução) e v_j ou não ficou atribuída ou ficou atribuída a um professor $p_{i'}$ que se situa abaixo de p_i na lista ordenada. Então, seria possível atribuir v_j a p_i retirando v_j a $p_{i'}$, pois $p_{i'}$ pode ficar colocado também em $v_{i'}$ sem qualquer restrição, e desse modo encontrar uma solução lexicograficamente menor e portanto melhor. Logo, *na fase de contratação, os emparelhamentos estáveis óptimos (do ponto de vista dos candidatos) correspondem às soluções lexicograficamente mínimas.*

Mostra-se agora que a função de peso proposta garante que os emparelhamentos de peso máximo correspondem às soluções lexicograficamente mínimas. A ideia subjacente à escolha dos pesos propostos foi a de evitar que candidatos mais graduados fossem ultrapassados nas suas preferências para se melhorar globalmente o valor (peso) das colocações dum grupo menos graduado. Ou seja, evitar que emparelhamentos instáveis pudessem ter peso máximo. Suponha-se que $(\beta_1, \ldots, \beta_n)$ e $(\beta_1', \ldots, \beta_n')$ são os valores das colocações de $p_1, \ldots, p_n$ em dois emparelhamentos. Pode-se mostrar que os pesos escolhidos garantem que,

$$\text{se } \beta_n + \beta_{n-1} + \ldots + \beta_{i+1} + \beta_i \geq \beta_n' + \beta_{n-1}' + \ldots + \beta_{i+1}' + \beta_i' \text{ então } \beta_i \geq \beta_i',$$

para todo $1 \leq i \leq n$. Atendendo à definição dos pesos, esta propriedade implica não só a estabilidade do emparelhamento, mas também que o emparelhamento de peso óptimo seja óptimo do ponto de vista dos candidatos. Para ver que a propriedade é válida, suponha-se que $\beta_i < \beta_i'$, e que

$$\beta_n + \beta_{n-1} + \ldots + \beta_{i+1} + \beta_i \geq \beta_n' + \beta_{n-1}' + \ldots + \beta_{i+1}' + \beta_i' \tag{1}$$

Ora, $\beta_i - \beta_i' = r_i(\gamma_i + 1 - t - (\gamma_i + 1 - t')) = r_i(t' - t)$ para t e t' iguais aos números de ordem na lista de preferências de p_i das posições atribuídas, de acordo com a definição de pesos introduzida. Logo, se $\beta_i - \beta_i' < 0$ então $t' - t \leq -1$. Então, para se ter (1) ou seja

$$\beta_n + \beta_{n-1} + \cdots + \beta_{i+1} \geq \beta_n' + \beta_{n-1}' + \cdots + \beta_{i+1}' + r_i(t - t')$$

[10]Gostaria de agradecer a Paulo Mateus (Dep. de Matemática & Centro de Lógica e Computação, Instituto Superior Técnico, Lisboa) esta observação.

[11]Neste contexto, a solução é um tuplo $(z_1, \ldots, z_n)$ dos números de ordem das posições atribuídas aos candidatos, segundo as listas de preferências respectivas.

teria que se ter também

$$(\beta_n + \beta_{n-1} + \cdots + \beta_{i+1}) - (\beta'_n + \beta'_{n-1} + \cdots + \beta'_{i+1}) \geq r_i$$

já que $t - t' \geq 1$. Mas,

$$\begin{aligned}
&(\beta_n + \beta_{n-1} + \cdots + \beta_{i+1}) - (\beta'_n + \beta'_{n-1} + \cdots + \beta'_{i+1}) \leq \\
&\leq \max(\beta'_n + \beta'_{n-1} + \cdots + \beta'_{i+1}) - \min(\beta_n + \beta_{n-1} + \cdots + \beta_{i+1}) \leq \\
&\leq (c(p_n, \Gamma_{n,1}) + c(p_{n-1}, \Gamma_{n-1,1}) + \cdots + c(p_{i+1}, \Gamma_{i+1,1})) - (1 + 1 + \cdots + 1) = \\
&= r_i - 1
\end{aligned}$$

chegando-se à contradição desejada. Por isso, se se verificar (1) então $\beta_i \geq \beta'_i$. Mas, para que $\beta_i \geq \beta'_i$, o número de ordem da posição que é atribuída a p_i na primeira solução não excede o número de ordem da posição que lhe fica atribuída pela segunda.

Mostrou-se assim que as soluções óptimas de $\textsc{MaxWeightPerfectMatching}(G)$

$$\begin{array}{|l}
\multicolumn{1}{l}{\textsc{MaxWeightPerfectMatching}(G)} \\
\text{maximizar } \sum_{e \in \mathcal{A}} c_e x_e \\
\text{sujeito a} \\
\quad \sum_{e \in \delta(i)} x_e = 1, \text{ para todo } i \in \mathcal{V}_1 \cup \mathcal{V}_2. \\
\quad x_e \in \{0, 1\}, \text{ para todo } e \in \mathcal{A}
\end{array}$$

para o grafo bipartido $G = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{A}, c)$ construído, correspondem às soluções lexicograficamente mínimas. Viu-se que estas soluções, na fase de contratação, determinam listas de colocações óptimas segundo os candidatos, que não violam a ordem pela qual os candidatos têm direito às posições. O candidato p_i fica colocado em v_j se $x_{(p_i, v_j)} = 1$ para algum j. Caso contrário, $x_{(p_i, v'_i)} = 1$ e p_i não fica colocado. A posição v_j fica atribuída ao candidato p_i se $x_{(p_i, v_j)} = 1$, para algum i. Caso contrário, $x_{(p'_j, v_j)} = 1$ e v_j fica livre.

5.2 Algoritmo para efectuar colocações em fases com destacamento

Quando existem candidatos que não podem ser colocados em vagas fictícias, como é o caso dos professores que estão a concorrer a destacamento, deixa de ser verdade que as soluções lexicograficamente mínimas sejam estáveis. O exemplo seguinte justifica esta afirmação.

Exemplo 10 *Três professores p_1, p_2 e p_3 concorrem a destacamento. Ocupando actualmente os lugares v_1, v_2 e v_3, respectivamente, concorrem com as listas de preferências seguintes (as quais estão totalmente ordenadas).*

p_1: v_3, v_2, v_1
p_2: v_1, v_2
p_3: v_1, v_3

Como as escolas não retêm os candidatos que pretendem deixar os lugares que nelas ocupam, a menos que estes não consigam outro lugar da sua preferência, a única solução estável aceitável é $\{(p_1, v_2), (p_2, v_1), (p_3, v_3)\}$. A solução que é lexicograficamente mínima é $\{(p_1, v_3), (p_2, v_2), (p_3, v_1)\}$ mas não é estável. De facto, nesta última, p_2 vê p_1 ocupar a vaga que pretendia, e desta forma, segundo o Decreto-Lei, teria sido ultrapassado por p_3.

De acordo com a função de peso introduzida acima, o emparelhamento de peso máximo seria $\{(p_1, v_3), (p_2, v_2), (p_3, v_1)\}$ e teria peso $9 + 1 + 2 = 12$, mas $\{(p_1, v_2), (p_2, v_1), (p_3, v_3)\}$ teria apenas peso $5 + 3 + 1 = 9$.

O próximo exemplo ilustra uma situação em que as soluções lexicograficamente mínimas são estáveis embora haja professores a concorrer a destacamento. Como se acabou de ver não se pode assumir que o sejam sempre e por isso, será proposto um novo método para tratar correctamente estes casos.

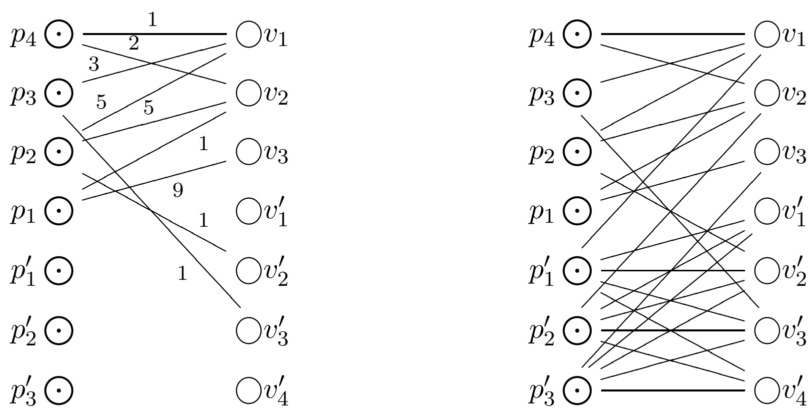
Exemplo 11 *Considere-se a seguinte situação, em que p_1 é o professor mais graduado e p_4 o menos graduado.*

$$\begin{array}{l} p_1\text{: } \{v_3\},\ \{v_2\} \\ p_2\text{: } \{v_1, v_2\},\ \{v_2'\} \\ p_3\text{: } \{v_1\},\ \{v_3'\} \\ p_4\text{: } \{v_2\},\ \{v_1\} \end{array}$$

Os pesos correspondentes seriam:

$$\begin{array}{ll} c(p_4, v_2) = 2,\ c(p_4, v_1) = 1, & r_4 = 1 \\ c(p_3, v_1) = 3,\ c(p_3, v_3') = 1, & r_3 = 2 - 1 + 1 = 2 \\ c(p_2, v_1) = 5,\ c(p_2, v_2) = 5,\ c(p_2, v_2') = 1, & r_2 = (2 + 3) - 2 + 1 = 4 \\ c(p_1, v_3) = 9,\ c(p_1, v_2) = 1, & r_1 = (2 + 3 + 5) - 3 + 1 = 8 \end{array}$$

O grafo que traduz o modelo introduzido encontra-se representado graficamente à direita. Os ramos omitidos no grafo dado à esquerda têm peso zero.



O emparelhamento óptimo seria $\{(p_1, v_3), (p_2, v_1), (p_3, v_3'), (p_4, v_2)\} \cup E'$ com, por exemplo, $E' = \{(p_1', v_1'), (p_2', v_2'), (p_3', v_4')\}$. Teria peso $9 + 5 + 1 + 2 = 17$ e corresponderia à sequência de números de ordem $(1, 1, 2, 1)$. Pode-se verificar facilmente que a solução é estável, neste caso.

Violar estabilidade para obter melhores colocações? É de referir também, que num trabalho recente, Sobrinho et al. [24] analisam casos em que se fosse violada a condição de estabilidade se conseguiria melhorar as posições atribuídas a alguns candidatos, como no Exemplo 12. Note-se que nesse trabalho se considerou que as listas de preferências dos candidatos estavam totalmente ordenadas, pelo que o problema tratado é diferente do que se aborda neste relatório.

Exemplo 12 . *Sejam p_1, p_2 e p_3 três candidatos, que actualmente ocupam v_1, v_2 e v_3 e concorrem a destacamento com as preferências seguintes.*

p_1: v_3, v_1
p_2: v_1, v_2
p_3: v_1, v_3

Não existe claramente nenhuma solução estável em que p_2 consegue o destacamento. Portanto, parecia razoável que, sem prejuízo de p_2, se admitisse $\{(p_1, v_3), (p_2, v_2), (p_3, v_1)\}$ como solução, ainda que este emparelhamento viole a estabilidade.

Dado que a condição de estabilidade é o único meio de que os candidatos dispõem para parcialmente verificarem que não estão a ser ultrapassados, a violação da estabilidade poderia suscitar várias reclamações e quiçá algumas desconfianças, que eventualmente não poderiam ser objecto duma resposta individual e convincente.

5.2.1 O problema geral como sequência de problemas de afectação clássicos

Para poder formular o problema geral da determinação duma lista de colocações óptima como um problema de afectação clássico seria necessário definir pesos que garantissem a estabilidade das soluções. Quando existem candidatos a concorrer a destacamento, isto talvez não seja possível.

Assim, vai-se alterar o modelo introduzido anteriormente, inserindo novos ramos no grafo que possibilitem a atribuição de v'_i (isto é, "não colocado") a qualquer candidato p_i mesmo quando p_i já detem uma posição anterior. Para tal define-se $G' = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{A}', c)$ com

$$\mathcal{A}' = \mathcal{A} \cup \{(p_i, v'_i) \mid \text{para todo } i \text{ com } 1 \leq i \leq n\}$$

o que quer dizer que, a solução de MaxWeightPerfectMatching(G') pode violar o Decreto-Lei n$^{\underline{\text{o}}}$ 35/2003. Por esta razão, se se verificar a existência de candidatos a destacamento que, de acordo com o emparelhamento de peso máximo obtido, ficariam agora por colocar, tem que se efectuar a resolução duma sequência de problemas do tipo MaxWeightPerfectMatching em vez dum único. Tais candidatos devem ser colocados definitivamente nas suas posições anteriores, e proceder-se-á depois à resolução do subproblema de colocação que resulta da sua remoção do modelo, bem como das suas vagas e dos nós fictícios correspondentes.

A função de peso. Quando se passa do modelo G ao modelo G' por inserção dos novos ramos (p_i, v'_i), para todo i, transforma-se um problema *com destacamentos* num problema de *contratação*. A lista de preferências de p_i passará a ser dada por

$$p_i : \begin{cases} \Gamma_{i,1}, \Gamma_{i,2}, \ldots, \Gamma_{i,\gamma_i}, \{v_j\}, \{v'_i\} & \text{se } v_j \text{ é a posição actual de } p_i \\ \\ \Gamma_{i,1}, \Gamma_{i,2}, \ldots, \Gamma_{i,\gamma_i}, \{v'_i\}, & \text{se } p_i \text{ não detem posição actualmente} \end{cases}$$

e designar-se-á por Ω_i o número de ordem do conjunto final, ou seja, o número de ordem da última preferência de p_i (a qual é agora v'_i). A função de peso que se propôs na Secção 5.1.1 pode ser usada agora também. Para os candidatos e vagas fictícias, $c(p'_j, v_j) = 0$, com $1 \leq j \leq k$ e $c(p'_j, v'_i) = 0$, com $1 \leq j \leq k$ e $1 \leq i \leq n$.

Como anteriormente, se $v \in \Gamma_{i,t}$ então t é o número de ordem da posição v nesta nova lista de preferências de p_i, isto é $o_{p_i}(v) = t$, para $1 \leq t \leq \Omega_i$. Atribui-se o mesmo peso $c(p_i, \Gamma_{i,t})$ a todas as posições que pertencem a um mesmo $\Gamma_{i,t}$, definindo-o por

$$c(p_i, \Gamma_{i,t}) = 1 + (\Omega_i - t) r_i$$

ou seja, $c(p_i, v) = 1 + (\Omega_i - t)r_i$, para todo $v \in \Gamma_{i,t}$ com $1 \leq t \leq \Omega_i$. Os valores $c(p_i, \Gamma_{i,t})$ são assim termos duma progressão aritmética de razão r_i e primeiro termo 1. O valor escolhido para a razão é, como anteriormente,

$$r_i = \begin{cases} 1 & \text{se } i = n \\ \sum_{i'=i+1}^{n} c(p_{i'}, \Gamma_{i',1}) - (n-i) + 1 & \text{com } 1 \leq i < n \end{cases}$$

o qual garante a estabilidade dos emparelhamentos perfeitos que têm peso máximo, quando todos os ramos de G' são aceitáveis. Se, no emparelhamento óptimo que se obtem para MaxWeightPerfectMatching(G'), nenhum candidato a destacamento p_i ficar na vaga fictícia v'_i correspondente, o processo pode terminar. Mas, se no emparelhamento óptimo obtido para MaxWeightPerfectMatching(G'), algum candidato p_i a concorrer a destacamento ficar colocado em v'_i, então esse emparelhamento não determina uma lista de colocações válida, pois viola a lei.

É importante notar que qualquer candidato p_i que ficar colocado em v'_i nalgum emparelhamento perfeito de peso óptimo, fica nessa posição em todos os emparelhamentos nessas condições. Conclui-se assim que tais candidatos só podem ficar colocados nas suas posições actuais. Consequentemente, todos os candidatos que ficarem colocados na sua última escolha ou seja, na posição v'_i, podem ser retirados do processo de colocação. Este será repetido sem esses candidatos e sem as suas posições actuais (sejam reais ou v'_i).

Podem ainda ser retirados todos os que, estando a concorrer a destacamento, ficarem colocados na sua posição actual. Depois destas remoções, não é necessário recomeçar todo o processo, mas apenas corrigir o emparelhamento imperfeito que se obtém desse óptimo quando se retiram tais nós e ramos. O Algoritmo de Kuhn-Munkres permite efectuá-lo.

Em resumo, as soluções óptimas de MaxWeightPerfectMatching(G') são estáveis e óptimas do ponto de vista dos candidatos para o modelo $G' = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{A}', c)$ proposto. Se o emparelhamento óptimo $E^\star$ for também um emparelhamento perfeito em G, então p_i fica na posição v_j sse $x_{(p_i,v_j)} = 1$ e não colocado sse $x_{(p_i,v'_i)} = 1$. Qualquer posição v_j tal que $x_{(p'_j,v_j)} = 1$ fica livre. Caso contrário, se $E^\star$ não for admissível em G, ou seja contiver ramos de $\mathcal{A}' \setminus \mathcal{A}$, então todo candidato p_i que ficar colocado numa vaga $v \in \Gamma_{i,t}$, para $t \geq \gamma_i + 1$, pode ser definitivamente colocado na sua "posição inicial", que é a que constitui $\Gamma_{\gamma_{i+1}}$. Então, p_i e a sua posição são retirados de G', juntamente com os nós fictícios correspondentes e os ramos incidentes nesses nós, e resolve-se o subproblema resultante. Não há necessidade de alterar a função de peso c. Depois desta remoção, o emparelhamento que resulta de $E^\star$ não é um emparelhamento perfeito no novo grafo, e não é possível completá-lo sem agravar (i.e., diminuir) o peso total.

Esse emparelhamento pode ser usado como solução de partida para a próxima iteração de colocações. Seria usado no Algoritmo de Kuhn-Munkres para construir um emparelhamento maximal com peso idêntico ao de $E^\star$. Tal emparelhamento não pode ser perfeito senão $E^\star$ não seria solução óptima para a iteração anterior. O Algoritmo de Kuhn-Munkres prosseguiria normalmente, efectuando a correcção de pesos e os passos habituais, até chegar a um novo emparelhamento perfeito (óptimo).

5.3 Pesos exponenciais mas complexidade polinomial

O número de problemas do tipo MaxWeightPerfectMatching(G') que será necessário resolver para encontrar uma lista de colocações óptima é o máximo entre um e o número de candidatos que estão a concorrer a destacamento.

Cada um dos problemas MaxWeightPerfectMatching(G') pode ser resolvido em $O((n+k)^4)$ pelo Algoritmo de Kuhn-Munkres (ou mesmo em $O((n+k)^3)$ se se usar estruturas convenientes para melhorar a eficiência da actualização de pesos) para n candidatos e k posições. No entanto esta expressão de complexidade requer que as comparações de pesos e a sua actualização possam ser feitas em tempo constante, o que não se pode garantir.

Para a função de peso proposta, as razões das progressões, e consequentemente os pesos, crescem exponencialmente, tornando-se $O((\max_i \gamma_i)^n)$. De facto, $\Omega_i \geq 2$, para todo i, e podemos facilmente verificar que se tem (2).

$$r_n = 1, \quad r_{i-1} = r_i \Omega_i = \prod_{q=i}^{n} \Omega_q \qquad \text{para todo } 1 < i \leq n \tag{2}$$

A função de peso cresce exponencialmente mesmo para subproblemas pequenos e por isso, é necessário escolher uma representação de pesos conveniente.

Passagem de MaxWeightPerfectMatching a MinWeightPerfectMatching. Para se ver mais facilmente que se pode tratar polinomialmente os pesos exponenciais, vai-se reduzir MaxWeightPerfectMatching(G') a MinWeightPerfectMatching(G') por transformação da função de peso c numa **função de custo** $\tilde{c}$,

$$\tilde{c}(p_i, \Gamma_{i,t}) \quad = \quad c(p_i, \Gamma_{i,1}) - c(p_i, \Gamma_{i,t}) = (t-1)r_i, \quad \text{para } 1 \leq t \leq \Omega_i \tag{3}$$

e substituição da função objectivo por *minimizar* $\sum_{e \in \mathcal{A}'} \tilde{c}_e x_e$.

Em cada iteração do Algoritmo de Kuhn-Munkers, é necessário identificar os ramos de $\mathcal{A}'$ que têm custo nulo, encontrar o arco que tem menor custo dentre os dum subconjunto particular de $\mathcal{A}'$ e ainda actualizar os custos dalguns ramos por adição ou subtracção desse valor mínimo de cada um. O Algoritmo de Kuhn-Munkers é um método primal-dual e estas actualizações de custo correspondem a alterações da solução do *dual*.

Dual MinWeightPerfectMatching(G')

$$\left| \begin{array}{l} \text{maximizar} \displaystyle\sum_{v \in \mathcal{V}_1 \cup \mathcal{V}_2} w_v \\ \text{sujeito a} \\ \qquad w_u + w_v \leq \tilde{c}_{uv}, \text{ para todo } (u,v) \in \mathcal{A}'. \end{array} \right.$$

Designe-se por $\overline{c}_e$ o custo actual do arco e depois duma ou mais alterações da solução dual. Inicialmente $\overline{c} = \tilde{c}$ e em todos os passos $\overline{c} \geq 0$. O trabalho no passo primal consiste na determinação dum emparelhamento de cardinal máximo num subgrafo de G' que tem por ramos

$$\mathcal{A}'^{=} = \{e \in \mathcal{A}' \mid \overline{c}_e = 0\}.$$

Seja $G'|_{\mathcal{A}'^{=}}$ tal grafo, isto é $G'|_{\mathcal{A}'^{=}} = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{A}'^{=}, \overline{c})$. Se esse emparelhamento de cardinalidade máxima for perfeito, então o método termina: é esse o emparelhamento de peso mínimo. Caso o emparelhamento não seja perfeito, haverá uma alteração de custos, que tem por base propriedades conhecidas, algumas das quais se referem agora.

Pelo teorema de König [6], em qualquer grafo bipartido, o número de ramos de qualquer emparelhamento de cardinalidade máxima é igual ao número de vértices de qualquer cobertura mínima do grafo. Uma cobertura (por vértices) é um conjunto de vértices tal que qualquer ramo do grafo é incidente nalgum vértice desse conjunto.

O Algoritmo de Kuhn-Munkers é muitas vezes descrito em termos das transformações que opera sobre a matriz de custos, com custos $+\infty$ para as entradas que não correspondem a ramos de $\mathcal{A}'$. Cada cobertura mínima de $G'|_{\mathcal{A}'^{=}}$ corresponde ao traçado dum número mínimo de rectas horizontais e/ou verticais que cubram todas as entradas nulas da matriz.

Quando $G'|_{\mathcal{A}'^{=}}$ não admite um emparelhamento perfeito, o número dessas rectas é inferior a $|\mathcal{V}_1|$ e, naturalmente inferior a $|\mathcal{V}_2|$, já que $|\mathcal{V}_1| = |\mathcal{V}_2|$). Nesse caso, pelo Algoritmo de Kuhn-Munkers, será necessário encontrar o mínimo das entradas que ficaram descobertas, subtraí-lo de cada uma dessas entradas não cobertas e somá-lo a cada uma das que ficaram cobertas por duas rectas. Seja μ_l esse mínimo na iteração l do Algoritmo de Kuhn-Munkers.

A actualização de custos pode ser também efectuada por subtracção de μ_l de cada coluna (ou, respectivamente, linha) que não ficou totalmente coberta e sua adição a cada linha (resp. coluna) que ficou coberta por uma recta horizontal (resp. vertical). O número total de rectas horizontais e verticais que foram traçadas para cobrir os zeros com um número mínimo de rectas na iteração l, digamos $R_l^H + R_l^V$, é menor do que $|\mathcal{V}_1|$. Os valores das funções de custos em duas iterações consecutivas l e $l+1$ estão relacionados por

$$\begin{aligned}\sum_{e\in\mathcal{A}'} \overline{c_{e,l}}\, x_e &= (|\mathcal{V}_1| - R_l^V)\mu_l - R_l^H\mu_l + \sum_{e\in\mathcal{A}'} \overline{c_{e,l+1}}\, x_e \\ &= (|\mathcal{V}_1| - R_l^H - R_l^V)\mu_l + \sum_{e\in\mathcal{A}'} \overline{c_{e,l+1}}\, x_e\end{aligned}$$

ou seja, $\sum_{e\in\mathcal{A}'} \overline{c_{e,l+1}}\, x_e$ é menor do que $\sum_{e\in\mathcal{A}'} \overline{c_{e,l}}\, x_e$, e os dois problemas de emparelhamento têm as mesmas soluções (embora valores diferentes da função objectivo). Aqui, $\overline{c_{e,l}}$ e $\overline{c_{e,l+1}}$ denotam os custos de e na iteração l e $l+1$.

O custo do emparelhamento de custo mínimo que se obtém quando o algoritmo termina é $\sum_{l=1}^{L-1}(|\mathcal{V}_1| - R_l^H - R_l^V)\mu_l$, em que L denota o número de iterações efectuadas até chegar a essa solução. Note-se que este custo óptimo é maior ou igual que $\sum_{l=1}^{L-1}\mu_l$. Este facto é importante no resto da prova. Tem-se (4)

$$\sum_{l=1}^{L-1}\mu_l \;\le\; \sum_{l=1}^{L-1}(|\mathcal{V}_1| - R_l^H - R_l^V)\mu_l \;\le\; r_1 - 1 \tag{4}$$

porque o emparelhamento perfeito de custo mínimo não pode ser pior do que a solução trivial $(1, \Omega_2, \ldots, \Omega_n)$, em que p_1 ficaria colocado numa das suas primeiras preferências e todos os restantes p_i's ficariam em v_i'. Sem perda de generalidade pode-se considerar que $\Gamma_{1,1} \neq \emptyset$. Se for vazio, deve ser retirado da lista de preferências de p_1 e deve-se procurar o primeiro $\Gamma_{1,t}$ não vazio. Se todos fossem vazios, bastaria retirar também p_1 e colocá-lo definitivamente na sua posição (actual ou de não colocação).

O custo determinado por $(1, \Omega_2, \ldots, \Omega_n)$ é $r_1 - 1$, porque é dado por $\sum_{i=2}^{n}(\Omega_i - 1)r_i$ e

$$\sum_{i=2}^{n}(\Omega_i - 1)r_i = \sum_{i=2}^{n}\Omega_i r_i - \sum_{i=2}^{n} r_i = \sum_{i=1}^{n-1} r_i - \sum_{i=2}^{n} r_i = r_1 - r_n = r_1 - 1$$

deduzindo-se a desigualdade (4).

Representar convenientemente os custos exponenciais. Conclui-se agora a prova de que as operações e comparações aritméticas se podem efectuar polinomialmente, de facto, linearmente em n, para cada subproblema, sendo n o número de candidatos a colocar (nesse subproblema). Para facilitar a demonstração, começa-se por supor que para resolver cada subproblema se re-inicializa o processo, não se partindo assim do emparelhamento obtido no passo anterior, recalculando-se $\overline{c}$ para o novo grafo reduzido. Represente-se também por $G' = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{A}', \overline{c})$ esse grafo. De (4) e (5)–(6)

$$r_n = 1 \quad r_{i-1} = r_i \Omega_i, \qquad \text{para todo } 1 < i \leq n \tag{5}$$

$$\tilde{c}(p_i, \Gamma_{i,t}) = (t-1) r_i, \qquad \text{para } 1 \leq t \leq \Omega_i \tag{6}$$

e do modo como os custos são actualizados, cada custo $\overline{c}_e$ pode-se escrever sempre como uma combinação linear de $r_1, \ldots, r_n$ com coeficientes inteiros não negativos $\alpha_1, \ldots, \alpha_n$ tais que $0 \leq \alpha_i < \Omega_i$, para $i \geq 1$, ou seja, que se tem (7).

$$\overline{c}_e \;=\; \sum_{i=1}^{n} \alpha_i r_i, \text{ com } \alpha_i \in \mathbb{Z},\, 0 \leq \alpha_i < \Omega_i, \text{ para } 1 \leq i \leq n, \text{ para todo } e \in \mathcal{A}' \tag{7}$$

De facto, $\sum_{l=1}^{L-1} \mu_l \leq r_1 - 1$ e a variação de $\overline{c}_e$ na iteração l é 0, μ_l ou $-\mu_l$. Se for $-\mu_l$ então $\overline{c}_e \geq \mu_l$ pois o mínimo μ_l das entradas que não ficam cobertas só é subtraído dessas entradas. Assim, os custos mantêm-se não negativos depois de cada transformação.

A variação total de $\overline{c}_e$ está limitada pela soma desses mínimos e, portanto, por $r_1 - 1$. Como o maior custo inicial é $(\Omega_1 - 1)r_1$, nenhum custo excederá $(\Omega_1 - 1)r_1 + (r_1 - 1)$, isto é, $\Omega_1 r_1 - 1$. Consequentemente, qualquer custo pode ser representado por uma combinação da forma (7), a qual fica univocamente determinada por $(\alpha_1, \ldots, \alpha_n)$.

> A *representação da soma* $(\alpha_1, \ldots, \alpha_n) + (\alpha'_1, \ldots, \alpha'_n)$ é um novo tuplo $(\alpha''_1, \ldots, \alpha''_n)$ que se obtem da forma habitual: processa-se as representações da direita para a esquerda, tomando $\tau := 0$ e definindo sucessivamente $\alpha''_i := (\tau + \alpha_i + \alpha'_i)\%\Omega_i$ $\tau := (\tau + \alpha_i + \alpha'_i)/\Omega_i$, para todo i, em que τ designa o transporte (ou *carry*) e $\%$ e $/$ o resto e quociente da divisão inteira. A operação preserva a condição $0 \leq \alpha_i, \alpha'_i, \alpha''_i < \Omega_i$, para todo i, e, de acordo com o exposto anteriormente, em qualquer soma efectuada pelo algoritmo, o último τ será 0.

> A *representação da diferença* $(\alpha_1, \ldots, \alpha_n) - (\alpha'_1, \ldots, \alpha'_n)$, que só será definida para $(\alpha_1, \ldots, \alpha_n) \geq_{LEX} (\alpha'_1, \ldots, \alpha'_n)$ é um novo tuplo $(\alpha''_1, \ldots, \alpha''_n)$ que também se obtem da forma habitual: processa-se as representações da direita para a esquerda, tomando $\tau := 0$ e definindo sucessivamente $\alpha''_i := (\alpha_i - \alpha'_i - \tau)$ e $\tau := 0$ se $\alpha_i - \alpha'_i - \tau \geq 0$ ou $\alpha''_i := (\Omega_i + \alpha_i - \alpha'_i - \tau)$ e $\tau := 1$ se $\alpha_i - \alpha'_i - \tau < 0$. É preservada a condição $0 \leq \alpha_i, \alpha'_i, \alpha''_i < \Omega_i$, para todo i, e o último τ será 0.

As representações são tratadas como representações de inteiros em bases de numeração, com excepção do facto dos intervalos de variação dos dígitos poderem ser distintos de posição para posição. É útil relembrar que $r_n = 1$ e $r_{i-1} = r_i \Omega_i$ para todo $1 < i \leq n$.

Em conclusão, as operações e comparações aritméticas, que o Algoritmo de Kuhn-Munkres realiza para resolver MinWeightPerfectMatching(G'), podem ser efectuadas em tempo e espaço linear no número de candidatos. Um custo $\overline{c}_e$ é nulo se todos os dígitos da sua representação forem nulos. A verificação de que $\overline{c}_e \leq \overline{c}_{e'}$ para $\overline{c}_e = \sum_{i=1}^{n} \alpha_i r_i$ e $\overline{c}_{e'} = \sum_{i=1}^{n} \alpha'_i r_i$ equivale à verificação de $(\alpha_1, \ldots, \alpha_n) \leq_{LEX} (\alpha'_1, \ldots, \alpha'_n)$. A determinação do valor mínimo das entradas não cobertas por rectas e actualização de custos, no passo dual, podem ser efectuadas em $O(n|A'|)$, ou seja, $O(n(n+k)^2)$.

Analise-se agora o caso em que se parte do emparelhamento obtido num dado subproblema e não se re-calcula $\overline{c}$. Observe-se que, qualquer lista de colocações nunca é pior do que aquela em que todos os candidatos que estão a concorrer a destacamento não o conseguem e todos os que não tinham posição anterior, continuam sem colocação (i.e., talvez desempregados). O custo desta solução é

$$\sum_{p_i \in \mathcal{I}_1} (\Omega_i - 2) r_i + \sum_{p_i \in \mathcal{I}_2} (\Omega_i - 1) r_i$$

em que $\mathcal{I}_1$ e $\mathcal{I}_2$ denotam os conjuntos de candidatos que estão e que não estão a concorrer a destacamento, respectivamente. Como,

$$\begin{aligned} \sum_{p_i \in \mathcal{I}_1} (\Omega_i - 2) r_i + \sum_{p_i \in \mathcal{I}_2} (\Omega_i - 1) r_i &\leq \sum_{p_i \in \mathcal{I}_2 \cup \mathcal{I}_1} (\Omega_i - 1) r_i = \Omega_1 r_1 - r_1 + \sum_{i=2}^{n} \Omega_i r_i - \sum_{i=2}^{n} r_i \\ &= \Omega_1 r_1 - r_1 + r_1 - r_n = \Omega_1 r_1 - 1 \end{aligned}$$

o custo óptimo não excede $\Omega_1 r_1 - 1$.

Seja $E^\star$ um emparelhamento de custo mínimo que se obtem por resolução de MinWeightPerfectMatching(G') e que viola a lei, por colocar numa posição fictícia algum dos candidatos que estavam a concorrer a destacamento. Como se viu anteriormente, deve-se então retirar todos os candidatos p_i que ficaram colocados nalgum $v \in \Gamma_{i,t}$ para $t \geq \gamma_i + 1$, retirar as vagas $\cup_{t \geq \gamma_i+1} \Gamma_{i,t}$ e os candidatos fictícios que lhe possam estar associados, para todo i, bem como os ramos incidentes nos nós retirados. Depois desta redução/transformação de G', o emparelhamento que resulta $E^\star$ não é perfeito, mas o Algoritmo de Kuhn-Munkres pode prosseguir normalmente, tomando o que resta de $E^\star$ como ponto de partida para a procura dum novo emparelhamento perfeito de custo óptimo (para o grafo reduzido). Se todos os ramos incidentes num dado nó tiverem custo positivo, então pode-se começar por subtrair a cada um deles o valor mínimo desses custos. Assim se garante que todos os nós têm ramos incidentes com custo zero. Note-se contudo, que se se deixar prosseguir o Algoritmo de Kuhn-Munkres normalmente, tal seria feito também (embora pudesse requerer algumas iterações). Designando por $\mathcal{S}$ a sequência de grafos G' para os quais se resolve MinWeightPerfectMatching(G') e por $L_{G'}$ o número de iterações realizadas pelo Algoritmo de Kuhn-Munkres nessa resolução, pode-se deduzir (8)

$$\sum_{G' \in \mathcal{S}} \sum_{l_{G'}=1}^{L_{G'}-1} \mu_{l_{G'}} \leq \sum_{p_i \in \mathcal{I}_1} (\Omega_i - 2) r_i + \sum_{p_i \in \mathcal{I}_2} (\Omega_i - 1) r_i = \Omega_1 r_1 - 1 \tag{8}$$

seguindo a argumentação da demonstração de (4). Inicialmente todos os ramos têm custo não negativo e menor ou igual que $(\Omega_1 - 1) r_1$. Logo, em todos os passos o valor de qualquer custo manter-se-á não negativo e não excederá $(\Omega_1 - 1) r_1 + \Omega_1 r_1 - 1$, ou seja $(2\Omega_1 - 1) r_1 - 1$. Continua a ser possível representar qualquer custo como um tuplo $(\alpha_1, \ldots, \alpha_n)$, mas agora com $0 \leq \alpha_i < \Omega_i$ para todo $i > 1$ e $0 \leq \alpha_1 < 2\Omega_1 - 1$. Conclui-se que todas as operações e comparações aritméticas que cada alteração de custos (passo dual) envolve, podem ser realizadas em $O(n(n+k)^2)$, também quando não se re-calcula a função de custo.

Para obter um emparelhamento perfeito pode ser necessário efectuar $O(|\mathcal{V}_1|^2)$ actualizações de custo (i.e., alterações da solução dual) [6], ou seja, $O((n+k)^2)$. Cada problema MinWeightPerfectMatching(G') pode ser resolvido em $O(n(n+k)^4)$, provando-se desta forma que o problema de colocações, com m candidatos a destacamento, pode ser resolvido em $O(\max\{1,m\}n(n+k)^4)$. A complexidade pode baixar para por exemplo, para $O(\max\{1,m\}n(n+k)^3)$, como se afirma na Proposição 1.

Proposição 1 *Seja n o número de candidatos total, m o número dos que estão a concorrer a destacamento e k o número de posições a concurso (que inclui as m posições dos candidatos a destacamento). Caso as listas de preferências dos candidatos contenham indiferença (empates) é possível ainda determinar uma lista de colocações óptima (segundo os candidatos) em $O(\max\{1,m\}n(n+k)^3)$.*

Com uma implementação mais cuidada do Algoritmo de Kuhn-Munkres pode-se realizar cada passo dual em $O(n(m+k))$. De facto, não há necessidade de actualizar um número quadrático de custos associados aos ramos mas apenas os $|\mathcal{V}_1|+|\mathcal{V}_2|$ componentes da solução dual. Por outro lado, pode-se manter informação associada aos vértices que permite determinar o valor mínimo μ_l em tempo linear, bem como a aresta correspondente e actualizar, se necessário, um número de custos apenas linear ([6], pags. 14 e 145-147).

5.4 Existência de vagas negativas

O modelo introduzido pode ser adaptado para contemplar também os casos em que há vagas negativas. Na situação extrema, as vagas negativas correspondem a posições, que só podem ficar ocupadas pelo seu actual detentor.

Em geral, as vagas negativas estão associadas a um conjunto de k' posições equivalentes, pretendendo-se, se possível, restringir o conjunto das que podem ficar atribuídas a um subconjunto próprio desse e cujo número de elementos não excede a diferença entre k' e o número de vagas negativas.

Seja $\{v_1,\ldots,v_{k'}\}$ um conjunto de posições equivalentes mas com $f \leq k'$ posições para abater (isto é, f vagas negativas). Ao modelo introduzido anteriormente, acrescente-se um conjunto de f candidatos $p''_1, \ldots, p''_f$, e ramos (p''_l, v_j) para todo $1 \leq l \leq f$ e $1 \leq j \leq k'$ e considere-se $c(p''_l, v_j) = +\infty$, em que $+\infty$ é um valor suficientemente grande. Acrescentem-se posições $v''_1,\ldots,v''_f$ e ramos (p''_l, v''_l) e (p'_i, v''_l) , para $1 \leq l \leq f$ e $1 \leq i \leq n$.

Cada ramo da forma (p'_i, v''_l) terá associado valor zero, e a sua ocorrência num emparelhamento corresponde ao abate de uma vaga negativa. O número de ramos do tipo (p''_l, v''_l) que ocorrerem num emparelhamento é igual ao número de vagas negativas que não foi possível abater. Também se poderia associar peso zero a cada um desses ramos, mas para calcular $+\infty$ da forma que a seguir se sugere, é mais simples não o fazer.

De facto, observe-se que para determinar $+\infty$, pode-se assumir que $p''_1, \ldots, p''_f$ ultrapassam qualquer outro candidato, e, sem perda de correcção, assumir-se também que têm idêntica graduação (e prioridade), e que cada candidato p''_l tem igual preferência por $\{v_1,\ldots,v_{k'}\}$ e estritamente inferior por v''_l. Pode-se então proceder como anteriormente, bastando usar um dos candidatos p''_l, para fixar o valor a atribuir a $+\infty$.

Esta adaptação do modelo permite concluir que instâncias do problema de colocação em que existem vagas negativas também podem ser resolvidas em tempo polinomial.

Exemplo 13 *Para ilustrar a ideia, considere-se a seguinte situação, em que se vai supor que v_2 (actual posição de p_1) é uma vaga negativa.*

$$
\begin{array}{l}
p_1\text{: } \{v_3\},\ \{v_2\} \\
p_2\text{: } \{v_1, v_2\},\ \{v_2'\} \\
p_3\text{: } \{v_1\},\ \{v_3'\} \\
p_4\text{: } \{v_2\},\ \{v_1\}
\end{array}
$$

Nesta representação, supõe-se que o candidato tem igual preferência pelas posições que se inclui num mesmo conjunto.

Acrescente-se então um candidato p_2'' e uma vaga v_2'', tendo p_2'' a seguinte lista de preferências.

$$p_2'' : \{v_2\}, \{v_2''\}$$

Os pesos correspondentes seriam

$$
\begin{array}{ll}
c(p_2'', v_2) = 17,\ c(p_2'', v_2'') = 1 & r_0 = 16 \\
c(p_1, v_3) = 9,\ c(p_1, v_2) = 1, & r_1 = (2+3+5) - 3 + 1 = 8 \\
c(p_2, v_1) = 5,\ c(p_2, v_2) = 5,\ c(p_2, v_2') = 1, & r_2 = (2+3) - 2 + 1 = 4 \\
c(p_3, v_1) = 3,\ c(p_3, v_3') = 1, & r_3 = 2 - 1 + 1 = 2 \\
c(p_4, v_2) = 2,\ c(p_4, v_1) = 1, & r_4 = 1
\end{array}
$$

e a lista ordenada passaria a ser $p_2'', p_1, p_2, p_3, p_4$. Pode-se verificar que a solução com maior valor é $\{(p_2'', v_2), (p_1, v_3), (p_2, v_2'), (p_3, v_3'), (p_4, v_1)\}$, pela qual, a vaga v_2 deixa de estar atribuída efectivamente.

O exemplo anterior mostra que a noção de *lista de colocações óptimas segundo os candidatos*, introduzida neste estudo, não viola o que a lei consigna (nomeadamente, no que diz respeito a vagas negativas) para melhor satisfazer os interesses dos candidatos.

5.5 Existência de posições equivalentes

Para simplificar o modelo e o processo de colocação, pode fazer sentido agrupar conjuntos de posições que são equivalentes, como, por exemplo, horários com a mesma carga horária ou lugares de quadro numa mesma escola, ou lugares em QZP, naturalmente, para um mesmo grupo e nível de docência (e fase do concurso).

Em ambas as situações tratadas nas secções anteriores, o número de variáveis de decisão do modelo pode ser reduzido se o problema passar a ser formulado como o da determinação dum fluxo máximo com peso máximo numa rede. No entanto, esse número continuará a ser da mesma ordem de grandeza do anterior, no pior dos casos.

6 Um Modelo Alternativo e Algoritmo de Optimização

O método de Kuhn-Munkres é um método primal-dual em que os pesos atribuídos aos ramos desempenham um papel essencial. Por outro lado, o número de variáveis de decisão do modelo matemático acima definido é $|\mathcal{A}|$, já que cada variável x_e está associada a um ramo $e \in \mathcal{A}$, e esse número pode ser demasiado grande nas instâncias reais[12]. Aliado ao facto de se conhecerem propriedades bastante fortes da estrutura dos emparelhamentos estáveis óptimos (i.e., das listas de colocações óptimas), é assim pertinente estudar modelos e algoritmos alternativos para resolução do problema que tirem partido dessas propriedades.

[12] Só a análise dos dados reais o permitirá concluir, mas face ao número de candidatos e ao conteúdo do artigo 12º do Decreto Lei nº 35/2003, de 27 Fevereiro (e republicação), ainda que se possam separar as colocações por grupos de docência (afins), são de prever valores de $|\mathcal{A}|$ talvez demasiado grandes.

Neste contexto, pode ser interessante estudar a aplicabilidade das ideias e métodos do paradigma de Programação por Restrições, o qual foi identificado pela *Association for Computing Machinery* (ACM) em 1996 como uma das áreas estratégicas de investigação em computação[13].

6.1 Programação por Restrições

O conceito de Programação por Restrições surgiu nas áreas de Inteligência Artificial e Computação Gráfica nos anos 1960s e 1970s, mas o desenvolvimento de investigação mais sistemática e activa sobre este tópico teve início nos finais dos anos 1980s. Em particular foi fulcral o trabalho de Jaffar e Lassez que estabeleceu os fundamentos teóricos das linguagens de Programação Lógica por Restrições [17]. Para uma introdução à área, consultar, por exemplo, [1, 3, 4, 7, 20, 22].

Uma ideia central em Programação por Restrições é a da redução dos domínios das variáveis de decisão do modelo por propagação de vários tipos de consistência local e por inferência de novas restrições durante a procura de soluções. Esta redução pode ser vista como uma forma de enumeração implícita de partes do espaço de procura. Os métodos de propagação de restrições funcionam como métodos incompletos de resolução, dado que, por questões de eficiência, só podem garantir parcialmente a consistência dos domínios das variáveis. Para obter soluções é quase sempre necessário combinar propagação de restrições e enumeração de valores para as variáveis.

Um *problema de satisfação de restrições em domínios finitos* (CSP) envolve um conjunto de variáveis de decisão $x_1, \ldots, x_n$ e um conjunto de restrições que especificam as combinações possíveis de valores dessas variáveis. Cada variável x_i está definida num domínio finito D_i, para $i = 1, \ldots, n$.

Gent et al. [10] e Gent e Prosser [11] apresentam codificações de StableMarriage e variantes como CSPs em domínios finitos e como SATs (i.e., problemas de satisfação de conjuntos de claúsulas do cálculo proposicional, em que, naturalmente, as variáveis são booleanas). Mostram nesses trabalhos que, para as instâncias de StableMarriage e StableMarriageWithIncompleteLists, essas codificações permitem enumerar todos os emparelhamentos estáveis sem falhas, quando se estabelece uma forma usual de consistência local, designada por consistência por arcos (*arc-consistency*). Um dos modelos baseia-se na utilização de matrizes de conflitos e o tamanho da codificação é $O(n^4)$, para n homens e n mulheres. O outro envolve $O(n^2)$ variáveis booleanas. Tornam-se assim algo complexos se se considerar a dimensão das instâncias do problema de colocação em estudo neste trabalho.

Os sistemas de Programação por Restrições disponibilizam linguagens de programação (que são também linguagens de modelação) e resolutores de restrições (que implementam métodos de propagação e de procura, tanto genéricos como específicos). Estes sistemas suportam vários tipos e formas de escrita das restrições, permitindo especificar modelos próximos da descrição matemática do problema. No contexto do problema de colocação, mais do que tentar simplesmente utilizar sistemas de Programação por Restrições, com os seus mecanismos genéricos de propagação, pode ser interessante desenvolver métodos que combinem técnicas de propagação de restrições implementadas nesses sistemas e propriedades específicas das soluções do problema.

[13] ACM Workshop on Strategic Directions in Computing, MIT, Cambridge, USA, June 1996. Disponível em `http://www.acm.org/pubs/surveys/sdcr/` (Março 2005).

6.2 Modelos de problemas de colocação como CSPs em domínios finitos

Como anteriormente, sejam $p_1, \ldots, p_n$ os candidatos (já ordenados), $v_1, \ldots, v_k$ as posições reais a concurso, $\Gamma_{i,1}, \ldots \Gamma_{i,\gamma_i}, \Gamma_{i,\gamma_i+1}$ a lista de preferências de p_i, ordenadas por ordem decrescente de preferência, mas com preferência idêntica por todas as posições que constam de cada $\Gamma_{i,t} \subseteq \{v_1, \ldots, v_k\}$, para $t = 1, \ldots, \gamma_i$. Supõe-se que $v_1, \ldots, v_k$ inclui também as posições dos candidatos já colocados que são opositores ao concurso.

Considerem-se variáveis de decisão y_i e z_i, para $i = 1, \ldots, n$, em que y_i indica a posição que p_i ocupará e z_i o número de ordem de y_i na lista de preferências de p_i, isto é, $z_i = o_{p_i}(y_i)$. Os domínios de y_i e de z_i, representados por $dom(y_i)$ e $dom(z_i)$, são dados por

$$\begin{array}{rcl} dom(y_i) & = & \cup_{t=1}^{\gamma_i+1} \Gamma_{i,t} \\ dom(z_i) & = & \{1, \ldots, \gamma_i + 1\} \end{array}$$

com $\Gamma_{i,\gamma_i+1} = \{v_j\}$, se p_i ocupava anteriormente a posição v_j e $\Gamma_{i,\gamma_i+1} = \{v_i'\}$, se não detinha qualquer posição. O problema de colocação pode ser então formulado pelo CSP em domínios finitos seguinte.

Modelo I

minimizar_LEX $(z_1, \ldots, z_n)$
sujeito a

$$\left\{ \begin{array}{l} y_i \in dom(y_i), i = 1, \ldots, n \\ z_i \in dom(z_i), i = 1, \ldots, n \\ z_i = o_{p_i}(y_i), i = 1, \ldots, n \\ y_i \neq y_{i'}, \text{para } 1 \leq i < i' \leq n \\ \text{se } y_{i'} \in dom(y_i) \text{ e } z_{i'} < \gamma_{i'} + 1 \text{ então } z_i \leq o_{p_i}(y_{i'}), \text{ para } 1 \leq i < i' \leq n \end{array} \right.$$

As duas últimas restrições impõem a não existência de candidatos distintos a ocupar uma mesma posição e a não violação da lista ordenada e o objectivo

$$\text{minimizar\_LEX } (z_1, \ldots, z_n)$$

é a determinação do tuplo $(z_1, \ldots, z_n)$ que é mínimo segundo a ordem lexicográfica e satisfaz as restrições definidas pelo modelo.

Posições equivalentes. Como atrás se referiu, pode ser útil agrupar algumas posições como equivalentes (por exemplo, os lugares dum mesmo grupo e nível de docência para uma escola, no concurso de provimento). O não agrupamento de posições equivalentes introduz uma complexidade desnecessária no Modelo I, pela existência de simetria (ainda que não explícita) de valores para as variáveis.

As posições $v_1, \ldots, v_k$ passam então a ser entendidas como posições não equivalentes, mas cada uma pode agora ser ocupada por mais do que um candidato: seja s_j o número máximo de candidatos que podem ser atribuídos a v_j, com $1 \leq j \leq k$. Por outro lado, não é necessário introduzir uma posição v_i' (i.e., não colocação de p_i) para cada candidato, dado que essas posições representariam um conjunto de posições equivalentes, sem limite máximo do número de candidatos que as podem ocupar. Em vez de $\Gamma_{i,\gamma_{i+1}} = \{v_i'\}$, defina-se apenas $\Gamma_{i,\gamma_{i+1}} = \{v'\}$, para todo p_i que não detem posição inicial. O modelo pode ser reformulado do seguinte modo.

Modelo II

$$
\begin{array}{l}
\text{minimizar\_LEX } (z_1, \ldots, z_n) \\
\text{sujeito a} \\
\left\{
\begin{array}{l}
y_i \in dom(y_i), i = 1, \ldots, n \\
z_i \in dom(z_i), i = 1, \ldots, n \\
z_i = o_{p_i}(y_i), i = 1, \ldots, n \\
|\{i \mid y_i = v_j\}| \leq s_j \text{ para } 1 \leq j \leq k \\
\text{se } y_{i'} \in dom(y_i) \text{ e } z_{i'} < \gamma_{i'} + 1 \text{ então } z_i \leq o_{p_i}(y_{i'}), \text{ para } 1 \leq i < i' \leq n
\end{array}
\right.
\end{array}
$$

Vagas negativas. O modelo anterior pode ser adaptado para tratar a existência de vagas negativas: seja s'_j o número de vagas negativas relativas a v_j, se as houver, para $1 \leq j \leq k$. O novo modelo terá que incluir restrições que impeçam a recuperação automática de s'_j posições do tipo v_j que sejam libertadas por candidatos anteriormente colocados. No entanto, tem que permitir que candidatos anteriormente colocados em v_j possam continuar a manter a sua colocação, se não conseguirem outra no concurso, mesmo que v_j se mantenha com vagas negativas. Introduza-se uma variável auxiliar w_j por cada v_j que tenha vagas negativas,

$$w_j = \max(0, |\{i \mid \Gamma_{i,\gamma_i+1} = \{v_j\} \text{ e } z_i \leq \gamma_i\}| - s'_j)$$

que indica o número de posições do tipo v_j que podem ser ocupadas por novos candidatos. Note-se que $z_i \leq \gamma_i$ indica que p_i consegue uma colocação diferente da sua actual posição v_j, pois $\Gamma_{i,\gamma_i+1} = \{v_j\}$. O modelo traduz-se então do modo seguinte.

Modelo III

$$
\begin{array}{l}
\text{minimizar\_LEX } (z_1, \ldots, z_n) \\
\text{sujeito a} \\
\left\{
\begin{array}{l}
y_i \in dom(y_i), i = 1, \ldots, n \\
z_i \in dom(z_i), i = 1, \ldots, n \\
z_i = o_{p_i}(y_i), i = 1, \ldots, n \\
|\{i \mid y_i = v_j\}| \leq s_j, \text{ para } 1 \leq j \leq k \text{ tal que } s_j > 0 \\
w_j = \max(0, |\{i \mid \Gamma_{i,\gamma_i+1} = \{v_j\} \text{ e } z_i \leq \gamma_i\}| - s'_j) \in \mathbb{Z}_0^+, \text{ para } 1 \leq j \leq k \text{ com } s'_j > 0 \\
|\{i \mid y_i = v_j \text{ e } z_i \leq \gamma_i\}| \leq w_j, \text{ para } 1 \leq j \leq k \text{ com } s'_j > 0 \\
\text{se } y_{i'} \in dom(y_i) \text{ e } z_{i'} < \gamma_{i'} + 1 \text{ então } z_i \leq o_{p_i}(y_{i'}), \text{ para } 1 \leq i < i' \leq n
\end{array}
\right.
\end{array}
$$

6.3 Codificação das restrições e ideias para procura de soluções

É importante referir que os sistemas de Programação por Restrições suportam restrições simbólicas que permitiriam codificar facilmente $z_i = o_{p_i}(y_i)$ e $|\{i \mid y_i = v_j\}| \leq s_j$, por exemplo. Neste caso seriam úteis as restrições de indexação com variáveis e de cardinalidade. No módulo de CLP(FD) para o SICStus Prolog [5, 23], por exemplo, são definidas por `element` e `count`[14].

[14] $z_i = o_{p_i}(y_i)$ poderia ser traduzida por `element(Ind,PrefsI,Yi), element(Ind,OPrefsI,Zi)`, em que `PrefsI` e `OPrefsI` representam o conjunto de preferências de p_i e o conjunto de números de ordem correspondentes. Para $|\{i \mid y_i = v_j\}| \leq s_j$ teria `count(VJ,Y,#=<,SJ)`, sendo `Y` é a lista das variáveis y_i, com $i = 1, \ldots, n$.

Esses sistemas disponibilizam formas do utilizador especificar novas restrições, bem como os seus modos de propagação, nomeadamente através da definição de *restrições globais*. Essas restrições podem estar em suspenso até ocorrer um evento que as torne activas e determine a sua propagação. São úteis para uma codificação eficiente das Restrições Estabilidade presentes nos três modelos anteriores.

Restrições Estabilidade:
se $y_{i'} \in dom(y_i)$ e $z_{i'} < \gamma_{i'} + 1$ então $z_i \leq o_{p_i}(y_{i'})$, para $1 \leq i < i' \leq n$

De facto, se durante a procura de soluções, o domínio da variável $y_{i'}$ ficasse reduzido a um valor, o que significaria que $y_{i'}$ só poderia tomar esse valor, a propagação dessa restrição a todos os z_i's ainda não instanciados, para $i < i'$, equivaleria à remoção de valores de $dom(z_i)$ que conduziriam a soluções instáveis.

Por outro lado, à semelhança do que acontece nos sistemas de Programação por Restrições, durante o processo de colocação pode-se inferir limites para $\min(z_i)$, os quais vão sendo actualizados pelas colocações que vão sendo efectuadas. Esses limites inferiores podem ser usados para propagar para a frente a restrição de que $y_{i'} \notin \cup_{1 \leq t < \min(z_i)} \Gamma_{i,t}$, para todo $i' > i$, a menos que se verifique $z_{i'} = \gamma_{i'} + 1$ (isto é, que $p_{i'}$ fique colocado na sua posição actual).

6.3.1 Vagas atribuídas seguindo lista ordenada e mantendo últimas preferências

Quando na colocação dos candidatos se procura primeiramente posição para o candidato que ainda não esteja colocado (provisoriamente) e se situe mais acima na lista ordenada, então, como acontece no Algoritmo de Gale-Shapley, a propagação de restrições determinadas pela colocação de p_i só afectará os candidatos $p_{i'}$ para $i' > i$, ou seja, só haverá necessidade de efectuar propagação para a frente.

Tanto na Extensão do Algoritmo de Gale-Shapley como no Algoritmo de Gale-Shapley orientado por internos, essa propagação para a frente é efectuada quando uma posição fica ocupada (ou é atingido o seu limite máximo de ocupantes, caso se trate duma posição múltipla), removendo da lista de aceitáveis os candidatos piores do que o que acabou de nela ser colocado. É também removida esta possível escolha da lista de preferências (ou seja, domínio) desses candidatos piores. No problema de colocação, esses são os candidatos que seguem p_i e que não detinham anteriormente posição y_i. Esta propagação não retira assim de $dom(y_{i'})$ a posição menos preferida de $p_{i'}$ (a última em que $p_{i'}$ pode ser colocado) e que é a que constitui $\Gamma_{i',\gamma_{i'}+1}$, qualquer que seja $i' > i$. Em particular, para os candidatos que anteriormente não detinham colocação, a posição v' (que significa que não ficará colocado) não será retirada do seu domínio por este processo de propagação, pois não tem limite máximo de vagas.

Quando um candidato p_i se propõe a v_j e v_j já está ocupada (ou já não tem vagas, se for entendida como um conjunto de posições equivalentes), então, se a referida propagação tiver sido efectuada, p_i tem direito a tal posição (porque a detinha anteriormente). Segundo as variantes do Algoritmo de Gale-Shapley referidas, p_i ainda constaria da lista de preferências de v_j, a qual necessariamente o aceitaria, rejeitando o pior candidato nela colocado provisoriamente: p_i ficaria definitivamente colocado nessa posição e o candidato rejeitado seria o próximo a procurar uma posição.

Pode-se pensar se haveria vantagem em implementar uma estratégia do tipo preguiçosa (*lazy*) para efectuar a propagação de restrições para a frente: apenas se p_i se propusesse a v_j e esta o rejeitasse, se retiraria v_j de $dom(y_{i'})$ para todo $i' > i$ tal que $\{v_j\} \neq \Gamma_{i',\gamma_{i'}+1}$.

Levando esta estratégia ao extremo, poder-se-ia não alterar os domínios iniciais das variáveis, pois o processo de rejeição garante que nenhuma das posições que violariam a estabilidade do emparelhamento seria atribuída.

No problema em estudo, as vantagens da propagação só surgem por, para cada valor de z_i, o conjunto Γ_{i,z_i} poder ser constituído por vários elementos (aqueles pelos quais p_i manifestou a mesma preferência). Nessa situação, para escolher o valor $v_j \in \Gamma_{i,z_i}$ que será atribuído primeiramente a y_i, pode ser interessante analisar as escolhas que restam para os candidatos que seguem i. Por outro lado, durante a procura duma *solução óptima*, a melhor solução $z^\star = (z_1^\star, \ldots, z_n^\star)$ que tiver sido encontrada anteriormente pode determinar o corte de caudas das listas de preferências dalguns candidatos, quando se impuser $z = (z_1, \ldots, z_n) <_{LEX} z^\star$. Nesse caso, a última escolha no novo domínio dum tal candidato $p_{i'}$, não se situando em $\Gamma_{i',\gamma_{i'}+1}$, obriga à propagação para trás de Restrições Estabilidade (se $y_{i'} \in dom(y_i)$ e $z_{i'} < \gamma_{i'} + 1$ então $z_i \leq o_{p_i}(y_{i'})$, para $1 \leq i < i' \leq n$) quando, por redução de $dom(y_{i'})$ a um elemento, se instanciar $y_{i'}$. Na Secção 6.3.3, aborda-se este ponto em detalhe.

6.3.2 Atribuição de posições por ordem arbitrária

A relação do problema de colocação com instâncias de variantes de StableMarriage permite perceber como determinar soluções estáveis sem efectuar as colocações dos candidatos pela ordem dada pela lista ordenada. O método de Gale-Shapley indica exactamente como resolver os conflitos que possam resultar.

Por analogia com o que se propõe para resolução de CSPs, na enumeração de valores para as variáveis podem ser tentadas outras ordens para a escolha da próxima variável a instanciar. Por exemplo, pode-se tentar analisar a aplicação de estratégias do tipo *first-fail principle*: instanciar primeiro variáveis cujos domínios sejam mais restritivos, o que, na versão mais elementar, corresponde a terem menor número de elementos.

No caso da existência de vagas negativas, pode ser interessante começar por procurar colocações para alguns dos candidatos que libertariam posições, em vez de, numa perspectiva optimista, começar por colocar um candidato mais graduado numa talvez vaga negativa, e esperar que essa vaga possa vir a ser tornada efectiva pela deslocação doutros candidatos.

Quando a escolha do próximo candidato p_i a colocar não segue a lista ordenada, começa a fazer sentido a propagação de restrições não só para baixo (isto é, para $i' > i$) mas também para cima (i.e., a $p_{i'}$ ainda não colocado, para $i' < i$). Da propagação para cima pode resultar a remoção de caudas das listas de preferências e os domínios acabarem por ficar vazios.

Como já se referiu na secção anterior, na procura de soluções óptimas, esta flexibilidade da ordem de escolha dos candidatos é importante. No entanto, em casos de insucesso, a recuperação em retrocesso deixa de poder seguir simplesmente o que é feito, por exemplo, no Algoritmo de Gale-Shapley (orientado por internos). É possível que o insucesso possa ser resolvido por deslocação doutros candidatos que tenham listas de preferência com empates.

Outra vantagem da relação estabelecida com StableMarriage é o entendimento de que é possível considerar que são as posições que se propõem aos candidatos, em vez de serem estes que se propõem. Como, em certas fases do concurso, o número de vagas tende a ser diminuto, é possível que procedendo deste modo se consiga melhorar a eficiência do método, na prática.

6.3.3 Determinação de listas de colocações óptimas

Em alternativa aos métodos polinomiais referidos na Secção 5, e atendendo a que as soluções procuradas são as que minimizam $(z_1, \ldots, z_n)$ segundo a ordem lexicográfica, pode-se tentar aplicar uma estratégia do tipo ramificar-limitar (*branch-and-bound*) para sua determinação. Note-se que o método não terá complexidade polinomial.

A ideia é efectuar uma pesquisa em árvore, com enumeração de valores para $z_1, \ldots, z_n$, por esta ordem, atribuindo a z_i valores do mínimo para o máximo do seu domínio, para $i = 1, \ldots, n$. Estes limites dos domínios vão sendo dinamicamente alterados por propagação de restrições impostas pelas colocações que se vão efectuando.

Quando no nó i se impuser $z_i = \min(z_i)$ ou $z_i \geq \min(z_i) + 1$, são criados dois ramos, sendo primeiramente explorado o da esquerda (isto é, $z_i = \min(z_i)$) e só depois o da direita, processo habitualmente designado por pesquisa em profundidade da esquerda para a direita.

Note-se que, caso seja encontrada alguma solução na sub-árvore esquerda, deixa de ser necessário expandir a sub-árvore direita, já que nenhuma solução que aí se venha a encontrar pode ser óptima. Se não for encontrada qualquer solução na sub-árvore esquerda, então pode-se propagar na sub-árvore direita Restrições Estabilidade que retirarão dos domínios de $y_{i'}$ e, possivelmente também de $z_{i'}$, posições que violariam a estabilidade da solução. Estas posições são os elementos do conjunto $\left(\cup_{t \leq \min(z_i)} \Gamma_{i,t}\right) \setminus \Gamma_{i',\gamma_{i'}+1}$, para $i' > i$.

Se, conforme anteriormente se descreveu, desta propagação resultar a redução do domínio duma variável y_l a uma só posição, então deve-se atribuir a essa variável esse valor. Deve-se propagar restrições aos candidatos anteriores $i' < l$ ainda não colocados e que estejam a concorrer também a essa posição, que garantam que o número de ordem da sua futura colocação $z_{i'}$ não excederá o número de ordem da referida posição na sua lista de preferências, isto é $o_{p_{i'}}(y_l)$.

Esta propagação para cima determina a redução dos domínios da variáveis nela envolvidas, os quais, eventualmente, podem ficar vazios, e determinar retrocesso (cronológico).

Quando, na pesquisa, se impõe $z_i = \min(z_i)$ e se avança para a descida na sub-árvore esquerda, o domínio de y_i fica reduzido a Γ_{i,z_i} ou um seu subconjunto. Se esse domínio for constituído por um único elemento, deve-se atribuir esse valor a y_i e propagar para baixo as restrições decorrentes disso.

Se o domínio de y_i não estiver reduzido a um elemento, então há que fixar uma ordem para escolha do primeiro valor a ser dado a y_i. À semelhança das abordagens de Programação por Restrições, pode-se tentar descobrir dinamicamente, um valor *melhor* do domínio de y_i, ou seja, um que possa ser causador de menos conflitos posteriormente (por exemplo, a posição menos requisitada dentre essas). Sendo v_j o valor escolhido, criam-se novamente dois ramos na pesquisa: $y_i = v_j$ ou $y_i \neq v_j$. Se se encontrar alguma solução na sub-árvore esquerda, pode-se propagar à sub-árvore direita a informação de que qualquer solução que nela venha a ser encontrada terá que ser melhor (isto é, $(z_1, \ldots, z_n)$ deverá ser lexicograficamente menor) do que a que a melhor que anteriormente se encontrou. Em particular, se z_l for a variável de menor índice ainda não instanciada quando se vai fixar $y_i = v_j$, então na sub-árvore direita, pode-se impor $z_l \leq z_l^\star$, em que $(z_1^\star, \ldots, z_n^\star)$ denota a melhor solução que já se encontrou. Para se propagar a restrição de que, na sub-árvore direita, $(z_1, \ldots, z_n) <_{LEX} (z_1^\star, \ldots, z_n^\star)$ pode-se seguir uma abordagem idêntica à das restrições globais, em Programação por Restrições.

Exemplo 14 *A Figura 1 apresenta a árvore de pesquisa para a instância*

p_1: $\{v_1, v_2, v_3\}$, $\{v'\}$
p_2: $\{v_1\}$, $\{v_2\}$, $\{v_3\}$
p_3: $\{v_4\}$, $\{v_2\}$
p_4: $\{v_1, v_4\}$, $\{v_2\}$, $\{v'\}$

em que se supõe que cada v_k só corresponde a uma vaga, para $k = 1, 2, 3, 4$ e que a posição v' não tem limite de vagas.

Finalmente, note-se que, embora não seja necessário, pode haver vantagem em iniciar este processo de pesquisa, partindo já duma primeira solução $(z_1^\star, \ldots, z_n^\star)$, que limitará toda a árvore. Uma tal solução pode ser determinada por aplicação do Algoritmo de Gale-Shapley (ou variantes), fixando uma ordem arbitrária para as posições, a qual implicitamente ordenará totalmente as listas de preferências dos candidatos (ou seja, também os $\Gamma_{i,t}$'s). Essa ordem deixa depois de ser considerada quando se passar à verificação da existência de soluções melhores, e à determinação duma melhor, se existir, por aplicação do processo de pesquisa em árvore descrito.

Do mesmo modo, pode haver vantagem em re-inicializar a pesquisa de cada vez que se consiga obter um melhoramento da solução, em vez prosseguir a procura por retrocesso cronológico.

Pode também haver vantagem em procurar melhorar primeiro as posições dos candidatos mais graduados. Observe-se que, com retrocesso cronológico, se tenta melhorar primeiro as posições dos menos graduados, dado que foram esses os últimos que foram colocados. Numa abordagem na linha da Programação Dinâmica, em vez de ter minimizar_LEX $(z_1, \ldots, z_n)$ como objectivo, pode haver vantagem em resolver sucessivamente "minimizar_LEX z_i sujeito a $z_1 = z_1^\star, \ldots, z_{i-1} = z_{i-1}^\star$, (e restantes restrições do modelo)", para $i \geq 1$, em que $(z_1^\star, \ldots, z_{i-1}^\star)$ é a melhor sequência encontrada na resolução de problemas análogos nos passos anteriores.

7 Observações Finais

Embora o Decreto-Lei nº 35/2003, de 27 Fevereiro (e republicação) não explicite que no preenchimento das vagas e dos horários serão procuradas listas de colocações óptimas, esperam-no de certo os opositores ao concurso, por ser esse o espírito da lei. Podendo as instâncias reais tornar-se demasiado complexas, surge a questão de saber se é sempre possível determinar listas de colocações óptimas (segundo os candidatos), dentro dos prazos previstos para o concurso, e quais as perspectivas de continuidade dessa possibilidade.

Como o número de posições a concurso tem vindo a diminuir, tal problema pode não se colocar, mas só a análise dos dados reais e as conclusões que dela se possam extrair, podem indicar se será oportuno ponderar uma revisão da lei. Uma possibilidade será, a bem da transparência, fixar no Decreto-Lei, critérios que regulem a ordenação das posições (horários ou lugares) existentes ou a surgir por recuperação de vagas, quer em concelhos quer em zonas pedagógicas. Os critérios serviriam depois para ordenar posições pelas quais os candidatos manifestaram igual preferência, de forma que as suas listas de preferências passem a estar totalmente ordenadas (sem empates). Deste modo, existirá, como se viu, uma única lista de colocações possível em cada fase do concurso, se se assumir que as escolas não vão tentar reter os seus docentes. Pelas razões apresentadas neste estudo, pode ser controverso efectuar uma ordenação aleatória já em fase de colocações, à semelhança, por exemplo, do

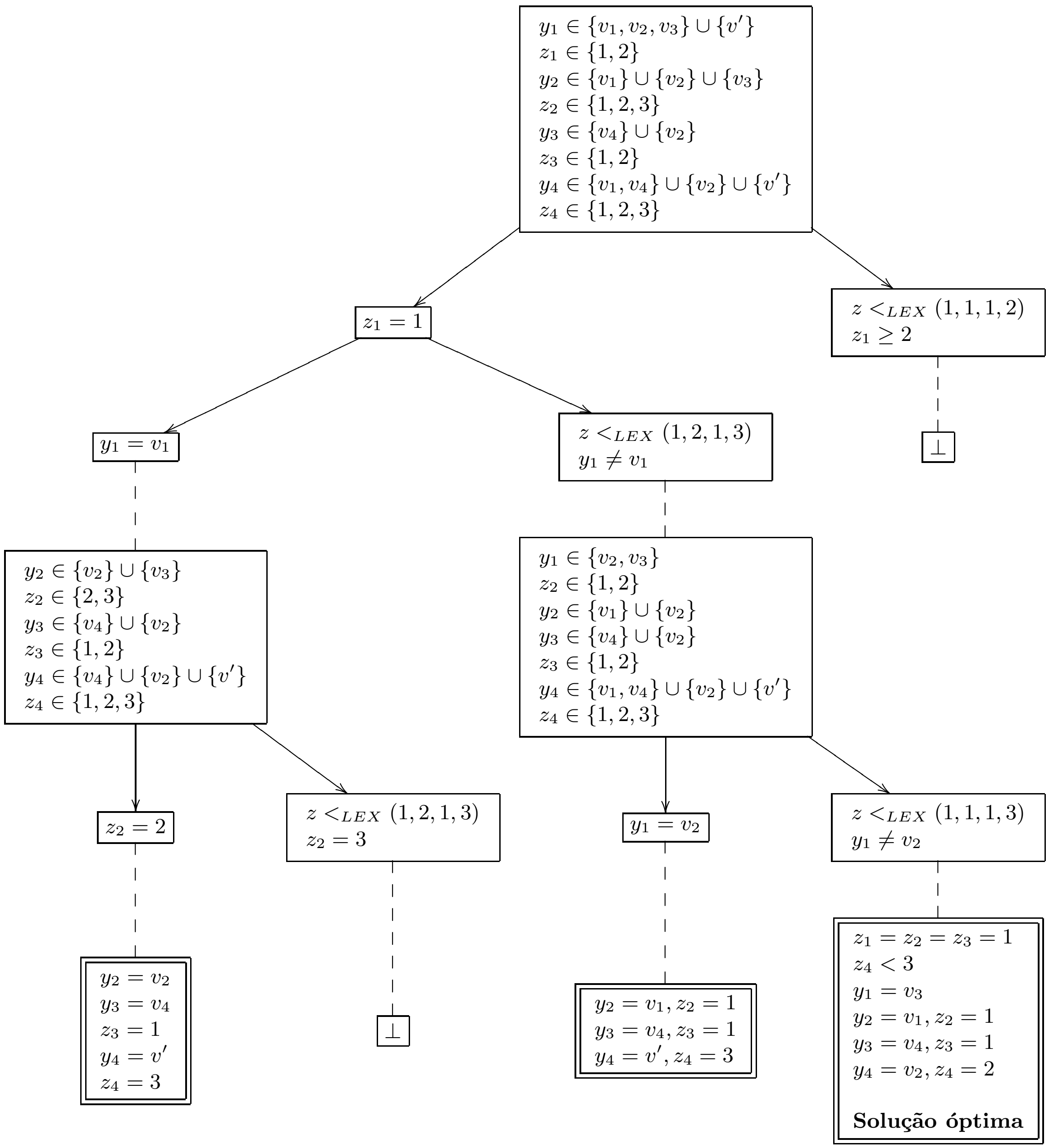


Figura 1: Árvore de pesquisa de lista de colocações óptimas. Os ramos a tracejado representam restrições deduzidas por propagação, considerando os números de ordem das preferências, e $\perp$ indica inconsistência.

que acontece no Scottish PRHO Allocations Scheme[15] (SPA) [19]. É de referir, no entanto, que este procedimento é, neste caso, do conhecimento dos intervenientes e o problema de optimização é NP-difícil, pois as listas de preferências (candidatos/hospitais) são realmente mútuas.

Agradecimentos

Gostaria de agradecer a António Leslie Bajuelos, o convite para apresentar este trabalho, ainda em fase de desenvolvimento, nos seminários do Centro de Estudos em Optimização e Controlo (CEOC) da Universidade de Aveiro em meados de Março, e a ajuda na localização do livro de Gusfield e Irving [13].

Muito agradeço também ao Professor Amilcar Sernadas o convite para apresentar o trabalho nos seminários do Centro de Lógica e Computação (CLC), Instituto Superior Técnico, Universidade de Lisboa, em meados de Abril, e a todos os colegas que me endereçaram os seus comentários, os quais resultaram num melhoramento e correcção da versão original deste relatório.

Expresso um agradecimento final aos alunos das licenciaturas em Ciência de Computadores e em Engenharia de Redes e Sistemas Informáticos e do mestrado em Informática do Departamento de Ciência de Computadores da Faculdade de Ciências da Universidade do Porto que realizaram com interesse e empenho os trabalhos que lhes foram propostos no âmbito das disciplinas do curso. E, em particular, aos alunos a quem propus a realização dum mini-projecto sobre este tema, no ano lectivo actual.

Referências

[1] K. R. Apt. *Principles of constraint programming*. Cambridge University Press, 2003.

[2] ATX Software. *Algoritmo de colocação de professores*. Novembro 2004.
`http://www.atxsoftware.com` (Março 2005)

[3] R. Barták. On-line Guide to Constraint Programming, 1998.
`http://kti.mff.cuni.cz/~bartak/constraints/` (Março 2005)

[4] R. Barták. Constraint Programming: In Pursuit of the Holy Grail. In *Proceedings of Week of Doctoral Students (WDS99)*, Part IV, MatFyzPress (1999) Prague, 555-564.
`http://www.visopt.com/Publications/WDS99.pdf` (Março 2005)

[5] M. Carlsson, G. Ottosson, B. Carlson. An open-ended finite domain constraint solver. In H. Glaser, P. Hartel, H. Kuchen (Eds), *Programming Languages: Implementations, Logics and Programs (PLILP'97)*, Lecture Notes in Computer Science **1292**, Springer-Verlag (1997) 191-206.

[6] W. Cook, W. Cunningham, W. Pulleyblank, A. Schrijver. *Combinatorial optimization*. John Wiley & Sons, 1998.

[7] R. Dechter. *Constraint processing*. Morgan Kaufmann, 2003.

[15] `http://www.dcs.gla.ac.uk/research/algorithms/stable/report.htm` (Março 2005)

[8] J. P. Faria. Estudo de casos – caso da colocação de professores. Acetatos da disciplina de "Engenharia de Requisitos de Sistemas de Software" do MEI e LEIC, Novembro 2004.

`http://paginas.fe.up.pt/~jpf/teach/ERSS/erss-coloca-profs.ppt` (Março 2005)

[9] D. Gale, L. S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly* **69** (1962) 9–15.

[10] I. P. Gent, R. Irving, D. F. Manlove, P. Prosser, B. Smith. A constraint programming approach to the stable marriage problem. In T. Walsh (Ed), *Principles and Practice of Constraint Programming - Proceedings of CP 2001*. Lecture Notes in Computer Science **2239**, Springer Verlag (2001) 225–239.

[11] I. P. Gent, P. Prosser. An empirical study of the stable marriage problem with ties and incomplete lists. In F. Harmelen (Ed) *Proceedings of the 15th Eureopean Conference on Artificial Intelligence, ECAI'2002*, IOS Press (2002) 141–145.

[12] D. Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM J. Computing* **16(1)** (1987) 111–128.

[13] D. Gusfield, R. W. Irving. *The stable marriage problem – structure and algorithms*. MIT Press, 1989.

[14] R. Irving. Stable marriage and indifference. *Discrete Applied Mathematics* **48** (1994) 261–272.

[15] K. Iwama, D. Manlove, S. Miyazaki, Y. Morita. Stable marriage with incomplete lists and ties. In J. Wiedermann, P. van Emde Boas, M. Nielsen (Eds) *Automata, Languages and Programming, 26th International Colloquium, ICALP'99*, Lecture Notes in Computer Science **1644**, Springer-Verlag (1999) 443–452.

[16] R. Irving, P. Leather. The complexity of counting stable marriages. *SIAM J. Computing* **15** (1986) 655–667.

[17] J. Jaffar, J.-L. Lassez. Constraint logic programming. In *Proc. ACM Symposium on Principles of Programming Languages (POPL'87)*. ACM (1987) 111–119.

[18] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quaterly* **2** (1955) 83–97.

[19] D. F. Manlove, R. Irving, K. Iwama, S. Miyazaki, Y. Morita. Hard variants of stable marriage. *Theoretical Computer Science* **276** (2002) 261–279.

[20] K. Marriott, P. Stuckey. *Programming with constraints*. MIT Press, 1998.

[21] J. Munkres. Algorithms for the assignment and transportation problems. *SIAM Journal on Applied Mathematics* **5** (1957) 32-38.

[22] S. Russell, P. Norvig. *Artificial Intelligence: A Modern Approach (Second Edition)*. Prentice Hall, 2002.

[23] SICStus Prolog User Manual (Release 3.12.0), SICS, Sweden, 2004.

[24] J. L. Sobrinho, J. B. Dias, P. Aguiar. O M(in)istério da Educação: ou o Problema da Colocação dos Docentes. (aceite para publicação pela Revista Ingenium, Março 2005)

http://www.sics.se/isl/sicstuswww/site/index.html (Março 2005)

[25] L. A. Wolsey. *Integer programming.* Wiley-Interscience, 1998.