Monografia de MAC5758
Um método híbrido para escalonar turnos de enfermeiras

Alexandre Albano
Marcio Oshiro

14 de dezembro de 2009

1 Introdução

Nesta monografia estudaremos o problema do escalonamento de turno de enfermeiras. Ou seja, o problema de em quais turnos cada enfermeira deve trabalhar.

Escalonamento de empregados a turnos de trabalho ou tarefas é um problema que tem sido estudado há bastante tempo. Do ponto de vista teórico, o caso particular das enfermeiras é bem interessante por haver uma quantidade muito grande de restrições envolvidas. Do ponto de vista prático é um problema com grande motivação, dado que as despesas com enfermeiras representam uma parcela expressiva do orçamento de um hospital. Além disso, um bom escalonamento pode melhorar a qualidade de vida das enfermeiras e conseqüentemente melhorar a qualidade no atendimento do hospital.

No artigo de Edmund Burk, Jingpeng Li e Rong Qu [BLQ10] é proposta uma técnica híbrida que combina programação inteira e uma meta-heurística chamada VNS (*Variable Neighbourhood Search*) para resolver esse problema de escalonamento. A idéia é tentar juntar as vantagens dos métodos exatos e métodos heurísticos já conhecidos. Dessa forma, espera-se obter soluções melhores com tempo de processamento menor. Esse método híbrido é comparado com uma solução comercial utilizada em um hospital holandês, revelando, de fato, melhores soluções.

2 O problema de escalonar enfermeiras

No problema do escalonamento de turno de enfermeiras o objetivo é atribuir, para cada enfermeira, um subconjuto de turnos de trabalho em um hospital por um determinado período de tempo (janela). Chamamos essa atribuição de escalonamento de enfermeiras. A tabela 1 mostra um exemplo de escalonamento de enfermeiras.

	4a-feira	5a-feira	6a-feira	sábado	domingo
Enfermeira 1	manhã	manhã	–	tarde	noite
Enfermeira 2	tarde	noite	noite	noite	tarde
Enfermeira 3	madrugada	tarde	manhã	–	–
Enfermeira 4	–	madrugada	madrugada	manhã	manhã

Tabela 1: Exemplo de escalonamento de enfermeiras

Certamente, não é qualquer escalonamento que nos interessa. O que procuramos é um escalonamento que satisfaça determinadas restrições. Tais escalonamentos são chamados de escalonamentos viáveis. No caso de enfermeiras, um exemplo de restrição desejada é a de cobertura, isto é, deve existir um número mínimo de enfermeiras trabalhando em cada turno de cada dia.

O planejamento de um bom escalonamento de enfermeiras é importante por vários motivos. Com um bom escalonamento podemos evitar a contratação excessiva de enfermeiras e garantir um número suficiente de enfermeiras no hospital. Dessa forma, diminui-se o gasto com contratações. Um bom escalonamento também pode evitar de sobrecarregar a carga de trabalho de uma enfermeira e tentar atribuir turnos de forma a melhorar a qualidade de vida e satisfação do trabalho das enfermeiras. Isso afeta diretamente a qualidade de atendimento de um hospital.

Encontrar um bom escalonamento de enfermeiras aparenta ser uma tarefa bem mais desafiadora do que o problema mais geral de escalonamento de empregados. A grande diferença está em algumas particularidades referente ao ambiente de trabalho das enfermeiras. Diferente de outros ambientes de trabalho, um hospital precisa estar sempre funcionando 24 horas por dia. Devido a isso, existem leis e acordos trabalhistas visando evitar um escalonamento que prejudique de alguma forma a saúde das enfermeiras. Em geral, por esse e outros motivos, um escalonamento de enfermeiras precisa satisfazer um conjunto grande de restrições.

As restrições para um escalonamento de enfermeiras são definidas por leis trabalhistas, regras do hospital e do sindicato, preferências das enfermeiras, etc. Logo, as restrições podem mudar dependendo do hospital e sua localização. Vamos dividir as restrições em duas classes: fortes e fracas. Uma restrição é forte se ela precisa ser obrigatoriamente satisfeita para que o escalonamento seja viável. Uma restrição é fraca se ela é desejável, mas não é necessário que seja satisfeita para que o escalonamento seja viável. Em geral, as restrições fortes são aquelas impostas por leis e por regras de hospitais e sindicatos. Já as restrições fracas são geralmente referentes às preferências das enfermeiras. Apesar de não serem obrigatórias, queremos satisfazer o maior número possível de restrições fracas.

Edmund Burke, Jingpeng Li e Rong Qu [BLQ10] estudaram um problema real de escalonamento de enfermeiras. Esse problema consiste em designar enfermeiras para turnos em uma unidade de terapia intensiva de um hospital holandês. São considerados 4 tipos de turnos: madrugada, manhã, tarde e noite. O escalonamento precisa cobrir uma janela de 5 semanas para 16 enfermeiras. As enfermeiras podem ter contratos de trabalho diferentes. Não consideramos as qualificações das enfermeiras, pois elas já são bem experientes e as enfermeiras em treinamento não são consideradas.

As restrições fortes, que devem sempre ser satisfeitas, são as seguintes.

Fo1: Cobertura mínima de serviço (não pode haver turno com falta de enfermeiras)

Fo2: Para cada dia, uma enfermeira não pode iniciar dois turnos.

Fo3: Cada enfermeira não pode trabalhar mais do que um determinado número de dias (dentro da janela). Esse número máximo de dias depende da enfermeira.

Fo4: Cada enfermeira não pode trabalhar em mais do que um determinado número de fins de semana (dentro da janela). Para o tamanho de janela que usaremos, este número será fixado em 3.

Fo5: Cada enfermeira não pode trabalhar em mais do que um determinado número de turnos de madrugada (dentro da janela). Para o tamanho de janela que usaremos, este número será fixado em 3.

Fo6: Se uma enfermeira receber um turno de madrugada, então ela obrigatoriamente recebe também um turno de madrugada no dia anterior ou no dia posterior. Isto é, para cada enfermeira, é proibido que esta receba um turno de madrugada "isolado".

Fo7: Após uma série de turnos de madrugada, é obrigatório que uma enfermeira receba dois dias livres de folga.

Fo8: Cada enfermeira não pode trabalhar em mais do que um determinado número de turnos de madrugada seguidos (dentro da janela). Esse número máximo é fixado para todas as enfermeiras.

Fo9: Cada enfermeira não pode trabalhar mais do que um determinado número de dias seguidos (dentro da janela). Esse número máximo é fixado para todas as enfermeiras.

Fo10: Existe uma enfermeira que não pode receber turnos noturnos

As restrições fracas estão listadas abaixo. É certo que em uma situação real, não podemos satisfazer todas elas, mas queremos satisfazer o maior número possível de restrições fracas sem violar alguma restrição forte.

Fr1: Ter fim de semana completo (uma enfermeira ou tem dois turnos no fim de semana, ou não tem nenhum).

Fr2: Evitar de atribuir um turno de trabalho entre dois dias de folga

Fr3: Ter um número mínimo de dias de folga após uma série de turnos.

Fr4: Ter um número máximo/mínimo de turnos de um tipo específico designados

Fr5: Ter um número máximo/mínimo de dias de trabalho por semana.

Fr6: Número máximo de dias consecutivos de trabalho para enfermeiras contratadas em regime de meio período.

Fr7: Evitar certas seqüências de turnos (por exemplo, um turno da tarde em um dia e de manhã no dia seguinte)

2.1 Abordagens conhecidas para o problema de escalonamento de enfermeiras

Segundo [Kar72], a maioria dos problemas de escalonamento de enfermeiras é NP-difícil, ou seja, sob a hipótese de que P $\neq$ NP, não existem algoritmos eficientes para resolver tais problemas. Ademais, são considerados por [TK82] como mais complicados do que o problema do caixeiro viajante.

Em geral, as técnicas a disposição para se atacar o problema podem ser classificadas como exatas ou (meta)heurísticas. Tradicionalmente, programação matemática é a ferramenta associada às técnicas exatas [Bea97, BFM96, BP05, WP72]. Apesar da vantagem de otimalidade garantida, essas abordagens apresentam dificuldades computacionais devido ao tamanho muito grande do espaço de busca. Em algumas propostas, as dimensões do problema são restritas para se obter um modelo simplificado. Contudo, isso leva a soluções que não são aplicáveis em situações reais em um hospital.

Em face dessa dificuldade, ao menos desde 1979 [SBW79] atenção foi dada ao uso de heurísticas para resolver problemas de escalonamento de enfermeiras. Uma metaheurística muito aplicada ao problema é a de algoritmos genéticos [AD00, AD04, EM99, KYY$^+$01]. Mas também foram aplicadas as seguintes metaheurísticas: *simulated annealing* [BJ93], busca tabu [BKS03], *variable neighborhood search* [BCPB04], algoritmos mimetizantes [BCCB01].

Contudo, metaheurísticas têm os seguintes reveses: não produzem garantidamente soluções ótimas, não reduzem garantidamente o espaço de busca e não têm em geral um critério de parada bem definido. Além disso, devido à maioria dos problemas de escalonamento de enfermeiras apresentar muitas restrições, em muitas vezes, a região viável é desconexa e em geral metaheurísticas apresentam dificuldades ao lidar com esse cenário.

Considerando as vantagens e desvantagens de ambas abordagens, no trabalho [BLQ10] é apresentada uma tentativa de integração entre as duas abordagens. A metaheurística envolvida nessa tentativa é a *variable neighborhood search.*

3 *Variable neighborhood search*

A metaheurística *variable neighborhood search*, ou simplesmente VNS, é relativamente recente e é uma variação da busca local. Na busca local tentamos achar uma solução melhor que a atual procurando em uma região restrita do espaço de soluções viáveis que contém a solução atual. Isso é repetido até que não consigamos melhorar a solução atual. Esse processo pode ser muito lento, dependendo da vizinhança em que consideramos para uma solução. Então, basicamente, a idéia utilizada na VNS é mudar o conceito de vizinhança entre cada busca.

De forma um pouco mais precisa, primeiro definimos um número finito k de sistemas de vizinhanças, $N_1, N_2, \ldots, N_k$. Um sistema de vizinhanças é algo que, dado uma solução, diz quais são as soluções que são consideradas vizinhas. Por exemplo, um sistema de vizinhanças pode se definido pela seguinte equivalência: duas soluções são vizinhas se e somente se diferem apenas na troca entre dois turnos de duas enfermeiras quaisquer. Para facilitar o entendimento, podemos considerar que os sistemas de vizinhanças são funções que recebem uma solução x e devolvem o conjunto $N_i(x)$ de soluções vizinhas de x.

Definidos os sistemas de vizinhanças, a VNS começa com uma solução viável x e um sistema inicial de vizinhanças N_1. Consideramos as soluções vizinhas a x, segundo N_1. Se não for encontrada uma vizinha de x que seja melhor, passa-se para o próximo sistema de vizinhanças N_2. Repetindo essa busca para x considerando N_2, se for encontrada uma solução y melhor do que x, então o sistema de vizinhanças usado na busca volta a ser N_1. Como o número de sistemas de vizinhanças é finito, o algoritmo termina quando nenhuma solução melhor é encontrada ao se buscar no último sistema N_k.

A princípio, na descrição da metaheurística VNS não há nenhuma exigência quanto a relação entre os vários sistemas de vizinhanças considerados. No entanto, dependendo do que queremos alguma relação entre os sistemas de vizinhanças será necessária. Por exemplo, se quisermos garantia de encontrar uma solução ótima global e não apenas local, precisamos que, para cada solução viável x, $N_1(x) \cup N_2(x), \ldots, \cup N_k(x)$ seja toda a região viável da instância do problema.

A ordem em que os sistemas de vizinhanças são considerados também não é pré-determinada, mas pode influenciar bastante na eficiência do algoritmo. No entanto, não existem estudos teóricos que comprovem que uma certa ordem é melhor que outra. Então, o método utilizado para se escolher a melhor ordem é através de experimentos.

Algoritmo 1 *Variable Neighbourhood Search*

defina um conjunto de estruturas de vizinhanças N_k, $k = 1, \cdots, k_{max}$
obtenha uma solução inicial x
$k \leftarrow 1$
enquanto $k \leq k_{max}$ **faça**
  faça busca na vizinhança N_k de x
  seja x' a melhor solução encontrada
  se $x' < x$ **então**
    $x \leftarrow x'$
    $k \leftarrow 1$
  senão
    $k \leftarrow k + 1$
  fim se
fim enquanto
devolva a melhor solução encontrada

3.1 Aplicação do *Variable Neighborhood Search* ao escalonamento de enfermeiras

Nesta seção, explicaremos como é feita a aplicação da metaheurística VNS para o problema de escalonamento de enfermeiras.

O VNS busca um escalonamento de enfermeiras que satisfaz todas as restrições fortes. No entanto, muito provavelmente não será possível satisfazer todas as restrições fracas. Então um dos objetivos do VNS será minimizar o número de restrições fracas violadas.

Para aplicar o VNS ao problema do escalonamento de enfermeiras, consideramos um modelo heurístico com multi-objetivos para o problema. Nesse modelo as restrições fracas são avaliadas em relação a um parâmetro de custo. Assim, cada violação de uma restrição é contabilizada como uma penalidade a função objetivo que queremos minimizar. Logo, minimizando essa função objetivo, minimizamos o número de restrições fracas violadas (ou a importância das restrições violadas, dependendo dos parâmetros de custo).

3.2 Modelagem heurística

No modelo heurístico com multi-objetivos para o problema de escalonamento de enfermeiras, consideramos os seguintes parâmetros.

- I é o conjunto de enfermeiras;
- I_1, I_2, I_3 são os subconjunto de enfermeiras que trabalham 20, 32 e 36 horas por semana respectivamente. Note que $I = I_1 \cup I_2 \cup I_3$;
- J é o conjunto de índices dos últimos dias de cada semana; dentro da janela considerada. Por exemplo $J = \{7, 14, 21, 28, 35\}$;
- K é o conjunto de tipos de turnos;
- K' é o conjunto de pares de turnos cuja sucessão são indesejadas;
- c_k é o limitante superior desejado para o número de atribuições consecutivas do tipo de turno k;
- g_t é o limitante superior desejado para o número de dias de trabalho para as enfermeiras em I_t;
- h_t é o limitante inferior desejado para o número de dias de trabalho para as enfermeiras em I_t.

As variáveis de decisão utilizadas são:

$$x_{ijk} \in \{0, 1\}, \qquad \forall i \in I, j \in \{1, 2, \ldots, 7|J|\}, k \in K.$$

A função de custo será a seguinte.

$$\min \bar{G}(x) = [\bar{g}_1(x), \bar{g}_2(x), \bar{g}_3(x), \bar{g}_4(x), \bar{g}_5(x), \bar{g}_6(x), \bar{g}_7(x), \bar{g}_8(x)]\,,$$

onde

$$\bar{g}_1(x) = \sum_{i \in I} \sum_{j \in J} \left| \sum_{k \in K} \left[x_{i(j-1)k} - x_{ijk} \right] \right|$$

$$\bar{g}_2(x) = \sum_{i \in I} \sum_{j=2}^{7|J|-1} \max \left\{ 0, \sum_{k \in K} \left[-x_{i(j-1)k} + x_{ijk} - x_{i(j+1)k} \right] \right\}$$

$$\bar{g}_3(x) = \sum_{i \in I} \sum_{j=2}^{7|J|-1} \max \left\{ 0, \sum_{k \in K} \left[x_{i(j-1)k} - x_{ijk} + x_{i(j+1)k} \right] - 1 \right\}$$

$$\bar{g}_4(x) = \sum_{i\in I}\sum_{r=1}^{7|J|-3}\sum_{k\in\{1,3\}} \max\left\{0, \sum_{j=r}^{r+3} x_{ijk} - c_k\right\}$$

$$\bar{g}_5(x) = \sum_{i\in I}\sum_{j=2}^{7|J|-1}\sum_{k\in\{1,3\}} \max\left\{0, -x_{i(j-1)k} + x_{ijk} - x_{i(j+1)k}\right\}$$

$$\bar{g}_6(x) = \sum_{t=1}^{3}\sum_{i\in I_t}\sum_{w=1}^{|J|}\left[\max\left\{0, \sum_{j=7w-6}^{7w}\sum_{k\in K} x_{ijk} - g_t\right\} + \max\left\{0, h_t - \sum_{j=7w-6}^{7w}\sum_{k\in K} x_{ijk}\right\}\right]$$

$$\bar{g}_7(x) = \sum_{i\in I_1}\sum_{r=1}^{7|J|-3} \max\left\{0, \sum_{j=r}^{r+3}\sum_{k\in K} x_{ijk} \; - \; 3\right\}$$

$$\bar{g}_8(x) = \sum_{i\in I}\sum_{j=1}^{7|J|-1}\sum_{(k_1,k_2)\in K'} \max\left\{0, x_{ijk_1} + x_{i(j+1)k_2} \; - \; 2\right\}$$

As restrições fortes também fazem parte do modelo, mas não as mostraremos aqui. Mostraremos a formulação delas mais adiante, na seção 4.1, quando descrevermos a formulação do problema como um programa inteiro.

3.3 Sistemas de vizinhanças

Uma parte fundamental para o VNS é a definição dos sistemas de vizinhanças.

No artigo [BLQ10], os sistemas são denotados $N_1, N_2, \ldots, N_k$. As N_1 vizinhas de uma solução x são as soluções que diferem de x na troca de (exatamente) um turno entre enfermeiras. As N_2 vizinhas de x são as soluções que diferem de x na troca de (exatamente) um bloco de dois turnos (consecutivos) entre enfermeiras (por exemplo, seria como se Maria e Joana trocassem entre si seus turnos de dois dias seguidos, por exemplo, dias 10 e 11. Ou seja, Maria passa a fazer o turno que Joana faria nos dias 10 e 11 e Joana passa a fazer o turno que Maria faria nos dias 10 e 11). Para o sistema N_3, vale uma afirmação análoga para blocos de 3 turnos consecutivos.

Vamos exemplificar esses sistemas com um diagrama (figura 1), onde é mostrado um intervalo de quatro dias e turnos recebidos por três enfermeiras. No diagrama, os períodos manhã, tarde, noite e madrugada estão representados com 'Man', 'T', 'N' e 'Mad', respectivamente. Além disso, o que está representado são soluções vizinhas obtidas ao se trocar turnos (ou blocos de) entre as enfermeiras 1 e 3 somente.

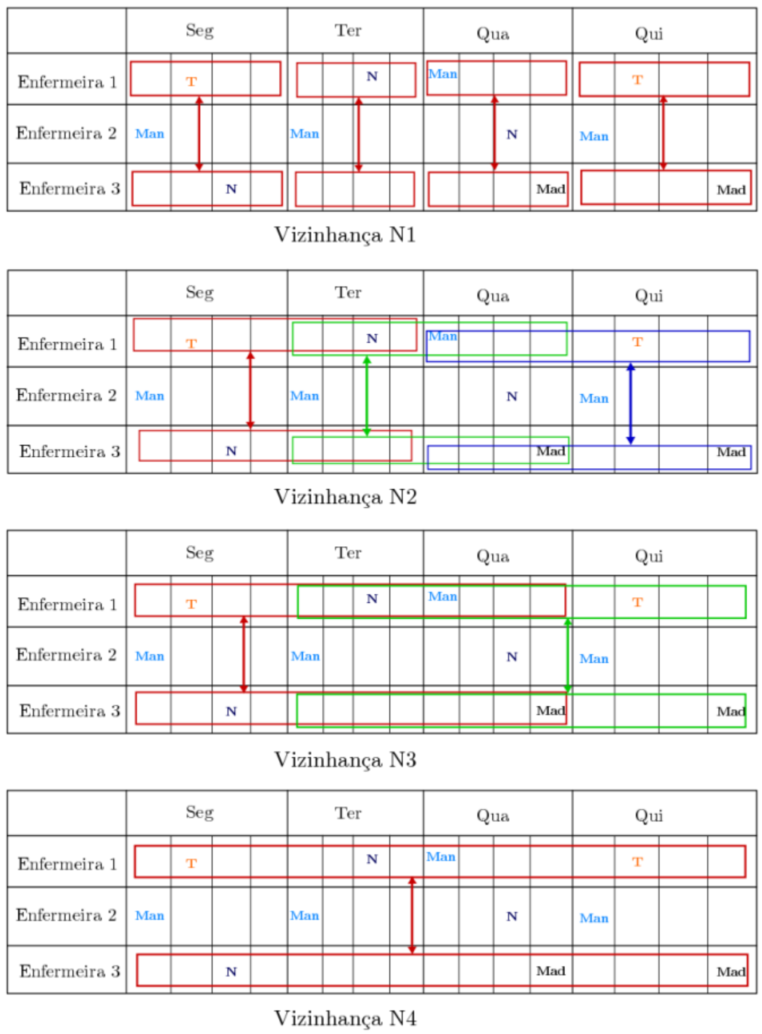

Figura 1: Diagrama ilustrando sistemas de vizinhanças

4 Métodos exatos

Existem métodos exatos para tratar o problema de escalonar enfermeiras, ou seja, métodos que encontram de fato um melhor escalonamento. Um método simples, mas bem ingênuo, seria enumerar todos os possíveis escalonamentos e escolher um melhor. No entanto, o número de escalonamento possíveis é exponencialmente grande, tornando a enumeração inviável.

Outro método exato bem mais razoável é através de programação matemática. Para isso, o problema é modelado como um programa inteiro e depois pode ser resolvido utilizando-se *softwares* específicos de programação matemática, como por exemplo o CPLEX e o Gurobi. Em seguida mostramos uma possível formulação do problema de escalonar enfermeiras. Vários parâmetros são utilizados para deixar o modelo mais flexível.

4.1 Modelo de programa inteiro

Assim como no modelo heurístico, consideramos um modelo multi-objetivo. Tanto as restrições fortes quanto as fracas aparecem no modelo. No entanto, as restrições fracas são formuladas com variáveis de folga, de forma que podem ser violadas e a violação seja detectada.

Os parâmetros utilizados são:

- I é o conjunto de enfermeiras;
- I_1, I_2, I_3 são os subconjunto de enfermeiras que trabalham 20, 32 e 36 horas por semana respectivamente. Note que $I = I_1 \cup I_2 \cup I_3$;
- J é o conjunto de índices dos últimos dias de cada semana; dentro da janela considerada. Por exemplo $J = \{7, 14, 21, 28, 35\}$;
- K é o conjunto de tipos de turnos;
- K' é o conjunto de pares de turnos cuja sucessão são indesejadas;
- d_{jk} é a cobertura necessária para o turno do tipo k no dia j;
- m_i é o número máximo de dias de trabalho dentro da janela para a enfermeira i;
- n_1 é o número máximo de turnos de madrugada consecutivos permitido dentro da janela;

- n_2 é o número máximo de dias de trabalho consecutivo dentro da janela;
- c_k é o limitante superior desejado para o número de atribuições consecutivas do tipo de turno k;
- g_t é o limitante superior desejado para o número de dias de trabalho para as enfermeiras em I_t;
- h_t é o limitante inferior desejado para o número de dias de trabalho para as enfermeiras em I_t.

As variáveis de decisão e de folga utilizadas são:

$$x_{ijk} \in \{0,1\}, \qquad \forall i \in I, j \in \{1,2,\ldots,7|J|\}, k \in K.$$

$$\begin{array}{rl}
s_{ij}^1 \geq 0, s_{ij}^2 \geq 0, & \forall i \in I, j \in J \\
s_{ij}^3 \geq 0, s_{ij}^4 \geq 0, & \forall i \in I, j \in \{2,3,\ldots,7|J|-1\} \\
s_{irk}^5 \geq 0, & \forall i \in I, r \in \{1,2\ldots,7|J|-3\}, k \in \{1,3\} \\
s_{ijk}^6 \geq 0, & \forall i \in I, j \in \{2,3,\ldots,7|J|-1\}, k \in \{1,3\} \\
s_{tiw}^7 \geq 0, s_{tiw}^8 \geq 0, & \forall t \in \{1,2,3\}, i \in I_t, w \in \{1,2,\ldots,|J|\} \\
s_{ir}^9 \geq 0, & \forall i \in I_1, r \in \{1,2,\ldots,7|J|-3\} \\
s_{ijk'}^{10} \geq 0, & \forall i \in I, j \in \{1,2,\ldots,7|J|-1\}, k' = (k_1,k_2) \in K'
\end{array}$$

A função de objetivo será a seguinte.

$$\min F(x) = \left[f_1(x)^T, f_2(x)^T, f_3(x)^T, f_4(x)^T, f_5(x)^T, f_6(x)^T, f_7(x)^T\right]^T,$$

onde

$$f_1(x) = [s_{ij}^1, s_{ij}^2]^T, \forall i \in I, j \in J$$
$$f_2(x) = [s_{ij}^3, s_{ij}^4]^T, \forall i \in I, j \in \{2,3,\ldots,7|J|-1\}$$
$$f_3(x) = [s_{irk}^5]^T, \forall i \in I, r \in \{1,2\ldots,7|J|-3\}, k \in \{1,3\}$$
$$f_4(x) = [s_{ijk}^6]^T, \forall i \in I, j \in \{2,3,\ldots,7|J|-1\}, k \in \{1,3\}$$
$$f_5(x) = [s_{tiw}^7, s_{tiw}^8]^T, \forall t \in \{1,2,3\}, i \in I_t, w \in \{1,2,\ldots,|J|\}$$
$$f_6(x) = [s_{ir}^9]^T, \forall i \in I_1, r \in \{1,2,\ldots,7|J|-3\}$$
$$f_7(x) = [s_{ijk'}^{10}]^T, \forall i \in I, j \in \{1,2,\ldots,7|J|-1\}, k' = (k_1,k_2) \in K'$$

5 Método híbrido

A grande vantagem do métodos exatos é que eles garantidamente sempre encontram uma solução ótima correspondente ao modelo utilizado. No entanto, a desvantagem está no alto consumo de tempo para se encontrar a solução ótima. Isso porque em geral são utilizados modelos de programação inteira e resolver programas inteiros é um problema NP-difícil.

Como na prática não dá para esperar tanto tempo para se obter uma solução, as metaheurísticas são muito utilizadas, apesar de não possuírem nenhuma garantia sobre a qualidade da solução devolvida. A vantagem das metaheurísticas é a velocidade.

Tentando equilibrar as vantagens e desvantagens dessas duas abordagens, Burke, Li e Qu [BLQ10] propõe um método híbrido. Dessa forma, espera-se encontrar soluções melhores com baixo consumo de tempo.

O método híbrido consiste em duas fases: na primeira fase, um método de otimização inteira é executado durante um tempo limitado. Na segunda fase, o *Variable Neighborhood Search* é executado, recebendo como solução inicial a melhor solução obtida na primeira fase. O VNS na segunda fase é utilizado para refinar a solução obtida pelo programa inteiro da primeira fase, ou seja, o VNS procura por soluções que satisfaçam mais restrições fracas nas vizinhanças da solução da primeira fase.

Como todas as restrições fortes e algumas fracas já foram tratadas na primeira fase, então os sistemas de vizinhanças para o VNS não precisam ser complexos. Burke, Li e Qu [BLQ10] utilizam, na segunda fase, os mesmos sistemas de vizinhanças descritos na seção 3.3.

6 Resultados experimentais

Os resultados são dados por um conjunto de testes, consistindo em execução por 49 minutos da primeira fase (exata) e 1 minuto da fase 2 (VNS). Todos os resultados apresentados a seguir foram retirados do artigo de Burke, Li e Qu [BLQ10].

O método híbrido foi comparado com dois métodos anteriores:

1. Algoritmo genético implementado em uma solução comercial da empresa ORTEC;

2. Algoritmo heurístico (VNS) implementado por Burke *et al.* [BCPB04].

Dados	Algoritmo Genético (após 1 hora)	VNS (após 1 hora)
Jan	775	735
Fev	1791	1866
Mar	2030	2010
Abr	612	457
Mai	2296	2161
Jun	9466	9291
Jul	781	481
Ago	4850	4880
Set	615	647
Out	736	665
Nov	2126	2030
Dez	625	520
Média	**2225**	**2145**

Tabela 2: Valores de soluções obtidas pelos métodos anteriores

A comparação é feita usando 12 instâncias reais, cada uma representando as demandas de um mês de um hospital holandês. O método híbrido é avaliado *versus* o melhor resultado obtido pelos 2 métodos anteriores citados acima. Na média, a melhora da qualidade da solução é de 15.2%, chegando até 46.3% em um determinado mês.

Em [BLQ10], explica-se o motivo da execução do método híbrido ter levado ao todo 50 minutos enquanto que os métodos anteriores executaram por 60 minutos: os testes realizados com os métodos anteriores utilizaram uma máquina com menor capacidade de processamento do que a utilizada para executar o método híbrido proposto.

Dados	Tamanho IP reduzido		Resultado IP		VNS do autor	
	Restrições	Variáveis	Após 49mins	Δ %	Após 1min	Δ %
Jan	7286	5995	631	14.1	460	37.4
Fev	6709	5588	1822	-1.7	1526	14.8
Mar	7203	5974	3890	-94	1713	14.8
Abr	6931	5760	1268	-177	391	14.4
Mai	7298	6067	5348	-147	2090	3.3
Jun	7044	5849	9126	1.8	8826	5.0
Jul	7170	5911	2498	-419	425	11.6
Ago	7362	6067	4582	5.5	3488	28.1
Set	6867	5708	680	-11.0	330	46.3
Out	7234	5963	605	9.0	445	33.1
Nov	7203	5974	2605	-28.0	1613	20.5
Dez	7106	5859	1037	-99.0	405	22.1
Média	**7118**	**5893**	**2841**	**-33.0**	**1809**	**15.2**

Tabela 3: Comparação de soluções do método híbrido

7 Conclusão

Foi argumentado e mostrado através de experimentos computacionais que um problema com grande número de restrições, como o problema de escalonar turnos de enfermeiras, pode ter boas soluções reveladas por um método híbrido, que inicialmente otimiza até um tempo limite um programa inteiro e após isso executa um algoritmo heurístico para aprimoramento da solução inicial.

O autor [BLQ10] pretende, em trabalhos futuros, explorar informação contida em soluções de um programa inteiro relaxado para guiar a subsequente busca local de soluções, e também acredita que a técnica de decompor as restrições do problema e levá-las em conta em métodos de otimização diferentes pode ter aplicações em uma classe ampla de problemas, como por exemplo problemas de alocação de recursos.

Referências

[AD00] Uwe Aickelin and Kathryn K. Dowsland. Exploring problem structure in a genetic algorithm approach to a nurse rostering problem. *Journal of Scheduling*, 3:139–153, 2000.

[AD04] Uwe Aickelin and Kathryn K. Dowsland. An indirect genetic algorithm for a nurse-scheduling problem. *Computers & Operations Research*, 31(5):761–778, 2004.

[BCCB01] Edmund K. Burke, Peter Cowling, Patrick De Causmaecker, and Greet Vanden Berghe. A memetic approach to the nurse rostering problem. *Applied Intelligence*, 15:199–214, 2001.

[BCPB04] Edmund K. Burke, Patrick De Causmaecker, Sanja Petrovic, and Greet Vanden Berghe. Variable neighborhood search for nurse rostering problems. In Mauricio G.C. Resende and Jorge Pinho de Souza, editors, *Metaheuristic: Computer Decision-Making*, pages 157–172. Kluwer Academic Publisher, 2004.

[Bea97] Nicholas Beaumont. Scheduling staff using mixed integer programming. *European Journal of Operational Research*, 98(3):473–484, 1997.

[BFM96] Ilham Berrada, Jacques A. Ferland, and Philippe Michelon. A multi-objective approach to nurse scheduling with both hard and soft constraints. *Socio-Economic Planning Sciences*, 30(3):183–193, 1996.

[BJ93] M.J. Brusco and L.W. Jacobs. A simulated annealing approach to the cyclic staff-scheduling problem. *Naval Research Logistics*, 40:69–84, 1993.

[BKS03] Edmund K. Burke, G. Kendall, and E. Soubeiga. A tabu-search hyperheuristic for timetabling and rostering. *Journal of Heuristics*, 9(6):451–470, 2003.

[BLQ10] Edmund K. Burke, Jingpeng Li, and Rong Qu. A hybrid model of integer programming and variable neighborhood search for highly-constrained nurse rostering problems. *European Journal of Operational Research*, 2010. (a ser publicado).

[BP05] Jonathan F. Bard and Hadi W. Purnomo. Preference scheduling for nurses using column generation. *European Journal of Operational Research*, 164(2):510–534, 2005.

[EM99] Fred F. Easton and Nashat Mansour. A distributed genetic algorithm for deterministic and stochastic labor scheduling problems. *European Journal of Operational Research*, 118(3):505–523, 1999.

[Kar72] Richard Manning Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Comuputer Computations*, pages 85–103. Plenum Press, 1972.

[KYY+01] H. Kawanaka, K. Yamamoto, T. Yoshikawa, T. Shingi, and S. Turuoka. Genetic algorithm with the constraints for nurse scheduling problem. In *Proceedings of Congress on Evolutionaru Computations*, pages 1123–1130, 2001.

[SBW79] L.D. Smith, D. Bird, and A. Wiggins. A computerized system to schedule nurses that recognizes staff preferences. *Hospital & Health Service Administration*, pages 19–35, 1979.

[TK82] James M. Tien and Angelica Kamiyama. On manpower scheduling algorithms. *Society of Industrial and Applied Mathematics*, 24:275–287, 1982.

[WP72] D. Michael Warner and Juan Prawda. A mathematical programming model for scheduling nursing personnel in a hospital. *Management Science*, 19:411–422, 1972.