

## META-HEURÍSTICA VNS APLICADA AO PROBLEMA DE ESCALONAMENTO DE ENFERMEIROS

**Leonardo Trento Volpato, Ademir Aparecido Constantino, Wesley Romão,  
Landir Saviniec**

Universidade Estadual de Maringá (UEM), Departamento de Informática  
Av. Colombo, 5.790, CEP 87020-900, Maringá-PR  
ltvolpato@gmail.com, ademir@din.uem.br, wesley.uem@gmail.com,  
landir.saviniec@gmail.com

### RESUMO

O problema de escalonamento de enfermeiros consiste em associar enfermeiros a turnos de trabalho diários maximizando a qualidade geral das escalas levando em consideração várias restrições. Motivados pela alta complexidade dos problemas de escalonamento de pessoal, muitos pesquisadores vem fazendo uso de meta-heurísticas na resolução dos mesmos. O presente trabalho propõe um novo algoritmo heurístico de duas fases baseado na resolução do Problema de Atribuição Multinível em conjunto com a meta-heurística VNS (*Variable Neighborhood Search*). O problema em questão possui restrições trabalhistas e preferencias pelos turnos previamente definidas. Na primeira fase uma solução inicial é gerada e, na segunda, são aplicados procedimentos de melhoramento. Os experimentos são executados com uma base de dados de quase 250.000 instâncias e os resultados são comparados com as melhores soluções da literatura. Os resultados indicaram uma redução significativa no custo médio das soluções e um aumento no número de soluções factíveis para alguns casos.

**PALAVRAS CHAVE.** Problema de Escalonamento de Enfermeiros, Meta-heurística, Problema de Atribuição, Heurística, Otimização Combinatória.

**Área principal:** PO na Área de Saúde, Meta-heurísticas, PO em Serviços.

### ABSTRACT

The nurse scheduling problem assigns nurses to shifts per day maximizing the overall quality of the roster while taking various constraints into account. Due to high complexity of personnel scheduling problems, many researchers have made use of meta-heuristics to solve them. This paper proposes a new two-phase heuristic algorithm based on resolution of the Multilevel Assigning Problem in conjunction with VNS meta-heuristic. This problem involves labor restrictions and shift preferences previously defined. In the first phase an initial is generated and in the second, improvement procedures are applied. The experiments are carried out on a benchmark dataset with almost 250,000 instances and the results are compared with the best solutions. The results indicate a significant average cost reduction and, for some cases, an increase of feasible solutions number.

**KEYWORDS.** Nurse Scheduling Problem. Meta-heuristic. Assignment Problem. Heuristic. Combinatorial Optimization.

**Main area:** OR in Health, Metaheuristics, OR in Services.

## 1. Introdução

A otimização de processos é uma necessidade comum para empresas, indústrias e instituições dos mais diversos tipos. Para resolver problemas de alta complexidade e comuns a alguns segmentos, técnicas computacionais da Pesquisa Operacional vêm sendo utilizadas com grande frequência. O Problema de Escalonamento de Pessoal (PEP) é um problema clássico de otimização e vem sendo abordado por pesquisadores a mais de 40 anos (Burke *et al.*, 2004a). De acordo com Ernst *et al.* (2004) o problema é classificado como NP-Difícil. Garey e Johnson (1979) classificam o Problema de Decisão associado como NP-Completo. Por este motivo, até o momento, não são conhecidos algoritmos exatos que resolvam tais problemas em tempo polinomial, o que incentiva muitos pesquisadores a investigarem novas abordagens que forneçam soluções de qualidade em um tempo computacional aceitável. Entre estas abordagens estão o uso de heurísticas e meta-heurísticas. Devido ao seu dinamismo, o processo de elaboração de escalas de trabalho se torna complexo e custoso, já que a demanda e horário de trabalho podem variar diariamente, por isso, a automatização desse processo pode trazer inúmeros benefícios, entre eles, redução de custos, melhoria na qualidade dos serviços, maior satisfação dos colaboradores e retenção de pessoal.

O Problema de Escalonamento de Enfermeiros (PEE), ou *Nurse Scheduling Problem* (NSP) é classificado como NP-Difícil (Osogami e Imai, 2000) e consiste em um Problema de Escalonamento de Pessoal que objetiva elaborar escalas de trabalho para equipes de enfermeiros respeitando uma grande variedade de restrições, entre elas, operacionais e trabalhistas. Além de atender as exigências comuns de uma escala de trabalho, o PEE busca atender também as preferências dos colaboradores de trabalhar em determinados turnos e/ou dias impactando diretamente na satisfação e motivação do pessoal. Levando em consideração os desafios do trabalho em ambiente hospitalar, a satisfação dos empregados torna-se um fator de extrema importância e imprescindível para a qualidade do serviço. Além disso, a redução de custos operacionais e o respeito às restrições do problema é de suma importância.

Este artigo segue o problema definido por Maenhout e Vanhoucke (2007). Neste caso as preferências dos enfermeiros por determinados turnos são previamente declaradas e são definidas associando um custo a cada possibilidade de turno para cada dia da escala. Custos também são utilizados para penalizar uma solução caso uma restrição não seja atendida, logo, o não atendimento de uma restrição implica na inclusão de uma penalidade ao custo da solução. Assim sendo, o PEE caracteriza-se como um problema de minimização dos custos associados às preferências dos enfermeiros pelos turnos e as penalidades aplicadas pelo desrespeito às restrições impostas. Dessa forma para um dado período  $D$  é escalado um conjunto  $N$  de enfermeiros, designando uma jornada para cada trabalhador. Os enfermeiros devem ser escalados para um dos  $S$  turnos possíveis em cada dia do período  $D$ .

Uma abordagem de resolução para o PEE utilizando a meta-heurística VNS foi proposta por Burke *et al.* (2004b) com uma base de dados da Universidade de Nottingham, do qual vários operadores de vizinhança são apresentados e combinados. Mesmo com a alta quantidade de restrições o algoritmo consegue facilmente fugir do ótimo local e gerar melhores soluções quando comparado a outras meta-heurísticas. Os experimentos mostraram que o uso de uma busca local intensiva reflete imediatamente na solução.

Bellanti *et al.* (2004) utilizou procedimentos de Busca Tabu e *Iterated Local Search* para a resolução de um PEE onde é necessário gerar escalas mensais de modo que atenda as restrições operacionais e trabalhistas impostas.

Um Algoritmo Genético (AG) foi proposto por Aickelin e Dowsland (2004) para o PEE em um hospital do Reino Unido. A abordagem utilizada é usar uma codificação indireta baseada em permutações entre os enfermeiros e um decodificador heurístico que constrói as escalas a partir dessas permutações.

Um algoritmo de Busca Dispersa foi proposto por Maenhout e Vanhoucke (2006). O método aplica um processo de intensificação de busca, mas mantém a diversificação da busca. Além da Busca Local (BL), um método iterativo também é aplicado para gerar soluções e efetuar uma série de recombinações para obter melhoramentos. O processo de recombinações efetua

alterações nas soluções iniciais que são conduzidas por soluções guias. As soluções resultantes podem ou não substituir membros do grupo das melhores soluções ou do conjunto das soluções que propiciam diversificação.

Ohki *et al.* (2006) utilizaram um AG cooperativo, onde cada indivíduo da população representa a jornada de um enfermeiro e a população toda representa toda a escala. Diferente de outras abordagens que utilizam o mesmo método na resolução do problema, esse algoritmo utiliza também o operador de mutação e não apenas o operador de cruzamento. As mutações são realizadas de modo que a validade das escalas não seja perdida e são responsáveis pela evasão de pequenos ótimos locais.

Maenhout e Vanhoucke (2007) propuseram um método de resolução que utiliza uma meta-heurística baseada na Lei de Coulomb denominada Meta-heurística Eletromagnética. O algoritmo deixa de atender a demanda para que as outras restrições sejam atendidas. Cada desobediência incorre no acréscimo de uma penalidade no custo da solução.

Gutjehra e Rauner (2007) apresentaram um algoritmo baseado na técnica ACO (*Ant Colony Optimization*) para a resolução do PEE. Os resultados das simulações mostraram que o ACO atingiu uma significativa melhora quando comparado a um algoritmo guloso proposto anteriormente e ao algoritmo Simulated Annealing.

Maenhout e Vanhoucke (2008) apresentaram um AG com diferentes operadores para a melhoria das soluções onde as soluções iniciais são construídas a partir da resolução de um Problema de Fluxo de Custo Mínimo para cada jornada. Assim que a solução inicial é construída, indivíduos são selecionados para que os operadores de cruzamento e mutação sejam aplicados.

Burke *et al.* (2008) propuseram um algoritmo heurístico híbrido que utiliza uma heurística de ordenação combinada com o método VNS. No processo de construção da solução inicial é utilizada uma ordenação heurística onde é possível distribuir as atribuições dos turnos de acordo com uma estimativa de custo de atribuí-las, ou qual a probabilidade desta ordenação implicar em penalidades de alto custo.

Tsai e Li (2009) propuseram um AG de dois estágios. No primeiro estágio, o algoritmo trata dos dias de folgas e dos dias trabalhados, por exemplo, tentando distribuir folgas aos feriados. No segundo estágio, o método efetua a distribuição e busca encontrar a melhor escala de turnos trabalhados.

Awadallah *et al.* (2011) apresentaram uma nova abordagem de resolução para o PEE, utilizando uma recente meta-heurística inspirada no processo de improvisação musical, conhecida como *Harmony Search Algorithm* (HSA) ou Algoritmo de Busca Harmônica. O método é baseado em população e utiliza conceitos de busca local. O funcionamento do algoritmo pode ser resumido da seguinte forma: primeiro, uma população inicial é gerada. Em seguida, a população é melhorada utilizando três decodificadores diferentes. A cada iteração apenas uma nova solução é gerada e substitui a pior solução da população. O processo é repetido até que um critério de parada seja atingido.

Constantino *et al.* (2013) apresentaram um método de duas fases baseado no problema de atribuição. O algoritmo, denominado MAPA, foi aplicado à base de dados NSPLib (Maenhout e Vanhoucke, 2007) e apresentou melhores resultados em relação a outros métodos que utilizam a mesma base de dados.

Neste artigo realiza-se uma extensão da abordagem proposta por Constantino *et al.* (2013), introduzindo um procedimento de busca local denominado *K-swap* e aplicando-se a meta-heurística VNS ao problema.

Este artigo está dividido em quatro seções. A seção 1 apresenta a introdução do problema e revisão da literatura. A seção 2 introduz o algoritmo construtivo, os algoritmos de busca local e, finalmente, o VNS. A seção 3 apresenta os resultados computacionais. As conclusões e trabalhos futuros são apresentados na seção de conclusões.

## 2. Algoritmo Proposto

O algoritmo proposto possui duas fases e combina dois métodos de busca local com o VNS. O algoritmo aborda o PEE como um Problema de Atribuição Multinível (PAM). O

problema é modelado como um grafo multipartido, sendo cada dia da escala representado por uma partição e cada atividade a ser realizada representada por um vértice. A solução é alcançada por meio de sucessivas resoluções de problemas de atribuição (PA), percorrendo a escala de trabalho e dividindo-a em partes que são recombinadas. Cada um dos problemas de atribuição é resolvido por um algoritmo fundamentado no algoritmo Húngaro, que garante solução do problema de atribuição em tempo polinomial. O PA objetiva encontrar uma permutação que relacione cada linha de uma matriz de custos a uma coluna específica, de modo que a soma dos custos de todas as designações seja a menor possível. Carpaneto e Toth (1987) modelam o PA da seguinte forma:

$$\text{Minimizar: } \sum_{i=1}^n \sum_{j=1}^n a_{ij} \cdot x_{ij} \quad (1)$$

$$\text{Sujeito a: } \sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (3)$$

$$x_{ij} \in \{0,1\}, i = 1, \dots, j = 1, \dots, n \quad (4)$$

Onde se considera:  $a_{ij}$  como custo de se atribuir a coluna  $j$  a linha  $i$ ;  $x_{ij}$  como a atribuição ou não da coluna  $j$  a linha  $i$ , assumindo 1 ou 0, respectivamente; e  $n$  representando a ordem da matriz de custos. A função-objetivo (1) deve ser minimizada. Nela, cada atribuição de uma coluna  $j$  a uma linha  $i$  é multiplicada pelo respectivo custo da atribuição. A restrição (2) determina que cada linha seja associada a uma única coluna. A restrição (3) obriga que cada coluna seja associada a uma única linha. A Equação (4) estabelece que a atribuição entre linhas e colunas seja representada por valores binários.

O algoritmo proposto neste trabalho possui duas fases e se baseia em resoluções sucessivas de PA de dois níveis. Na primeira fase uma solução inicial é gerada e na segunda fase procedimentos são aplicados em busca de melhorias na solução. Para isso, foi implementado o algoritmo de Nedas (2005) baseado no Algoritmo Húngaro (Kuhn, 2005) que encontra a solução ótima do PA sob complexidade assintótica  $O(n^3)$ .

## 2.1. Fase Construtiva

Na fase de construção da solução inicial um problema de atribuição é gerado e resolvido para cada dia da escala, sendo que cada problema é representado numa matriz de custos quadrada  $C=[c_{ik}]$  de ordem  $n$ , onde  $n$  é o número total de enfermeiros e cada elemento  $c_{ik}$  associa o custo do enfermeiro  $i$  receber, num dado dia  $j$ , uma dada tarefa de índice  $k$ . Esse custo associado a cada elemento é obtido somando-se o custo de preferência do enfermeiro  $i$ , no dia  $j$ , pela tarefa  $k$  ( $cp(i,j,k)$ ); o número de violações às restrições rígidas que tal tarefa acrescenta ( $nVRR$ ), multiplicado pela penalidade de cada violação à restrição rígida ( $PenVRR$ ); e o número de violações às restrições flexíveis incluídas pela respectiva tarefa ( $nVRF$ ), multiplicado pela penalidade de cada violação à restrição flexível ( $PenVRF$ ), tal qual mostra a função abaixo:

$$f(i, j, k) = cp(i, j, k) + PenVRR.nVRR + PenVRF.nVRF$$

Como nas instâncias utilizadas para a realização dos experimentos o número de enfermeiros é sempre maior ou igual ao número de tarefas exigidas pela demanda, sempre que necessário a matriz  $C$  é completada com tarefas fictícias até que o número de colunas, representando as tarefas seja igual ao número de enfermeiros, representado pelas linhas, permitindo assim a resolução do problema de atribuição. A estrutura da matriz de custos  $C$  é dada pela Figura 1.

	Tarefas	Tarefas Fictícias
Enfermeiros	$c_{ik}=f(i,j,k)$	$c_{ik}=\text{Mín } f(i,j,k);$ $k=1,\dots,s$

**Figura 1: Matriz de custos para os problemas de atribuição na fase de construção.**

No início, todas as posições das jornadas estão vagas. Para escalonar uma tarefa para cada enfermeiro no primeiro dia, a matriz  $C$  correspondente ao primeiro nível é gerada e resolvida pelo PA. Em seguida, as matrizes dos dias subsequentes também são geradas e os problemas de atribuição correspondentes são resolvidos. Caso alguma das tarefas atribuídas viole alguma das restrições impostas, a função  $f(i,j,k)$  inclui uma penalidade ao custo da solução. Ao final do processo todos os níveis são resolvidos e todas as jornadas estão completas. Abaixo, uma visão geral da fase construtiva:

**Início**

Inicialize os dados;

**Para  $j=1$  até  $d$  faça:**

Gere a matriz de custos  $C$  correspondente ao dia  $j$ ;

Resolva o PA da matriz  $C$ ;

Aloque as tarefas aos enfermeiros conforme o resultado obtido;

**Fim.**

## 2.2. Fase de Melhoramento

O algoritmo proposto utiliza a meta-heurística VNS e combina dois procedimentos de busca local. A meta-heurística VNS explora o espaço de soluções através de trocas sistemáticas de vizinhança. Partindo de uma solução inicial, o método seleciona um procedimento de busca local que é aplicado à solução. O conjunto de todas as soluções geradas na aplicação de um determinado procedimento de busca local define a estrutura de vizinhança daquele procedimento. Cada vez que um procedimento é aplicado e uma nova solução é gerada, esta é comparada com a solução corrente e a melhor solução dentre as duas se torna a solução corrente e então, outra busca local é realizada com o objetivo de se obter uma nova solução que será comparada com a melhor solução encontrada até o momento. Dessa maneira, o VNS segue até que algum critério de parada seja satisfeito. Os dois procedimentos de busca local considerados nesta abordagem são o  $K$ -Swap e o PCR (Procedimento de Cortes e Recombinações). As estruturas de vizinhança utilizadas são: PCR,  $1$ -Swap,  $2$ -Swap, ...,  $(d-1)$ -Swap, onde  $d$  é o número de dias da escala.

O  $K$ -Swap investiga as possibilidades de trocas de tarefas entre os enfermeiros dentro de uma escala parcial de tamanho  $k$ , variando desde as tarefas de um único dia até todas as tarefas da escala. Assim, para cada possível jornada parcial de tamanho  $k$ , é criada uma matriz de custos  $D=[d_{ij}]$ , onde cada elemento  $d_{ij}$  é obtido calculando-se o custo de atribuir ao enfermeiro  $i$  cada uma das tarefas da escala parcialmente designada ao enfermeiro  $j$ , custo calculado como na fase construtiva, somando-se os custos de preferências do enfermeiro e os custos referentes às violações das restrições oriundas da substituição de cada uma das tarefas daquela jornada parcial em sua jornada já existente. O  $K$ -Swap percorre toda a escala redistribuindo as tarefas de todos os dias onde uma iteração do procedimento corresponde a  $k$  redistribuições, por isso, o nome  $K$ -

Swap. A Figura 2 ilustra as possíveis recombinações para o quarto dia da jornada (1-Swap).

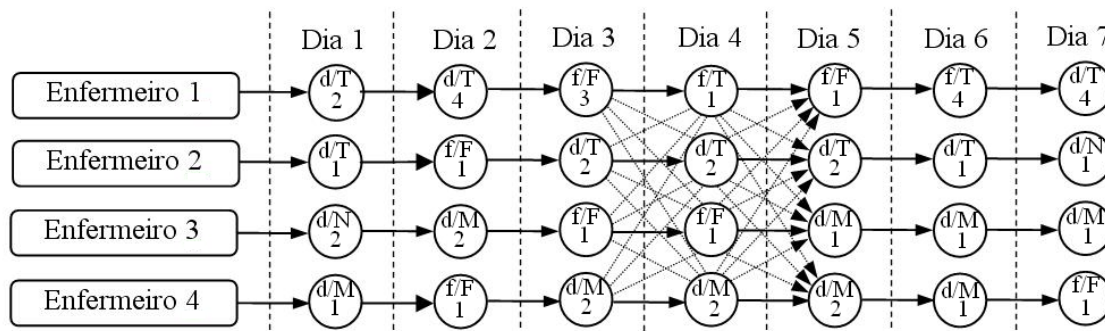


Figura 2: Solução inicial e possíveis recombinações utilizando o 1-Swap (para apenas um intervalo de tempo) aplicado ao quarto dia da escala.

A Figura 3 ilustra um exemplo do 2-Swap envolvendo os blocos do quarto e quinto dias.

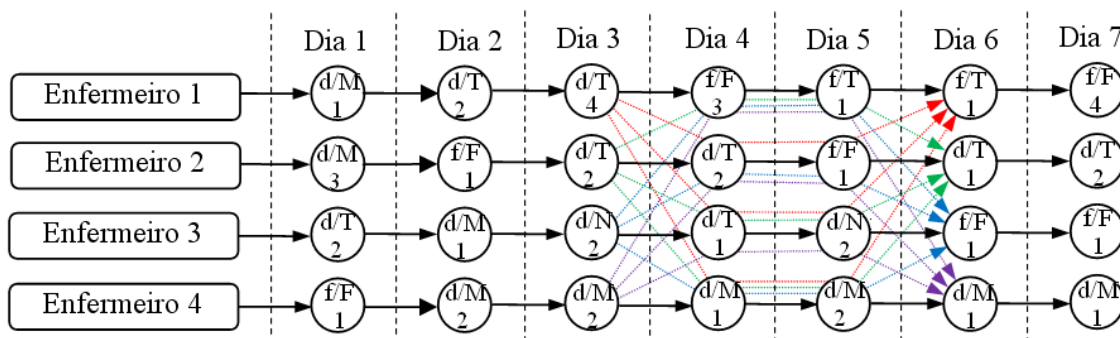


Figura 3. Ilustração de 2-Swap considerando dois intervalos de tempo consecutivos, aplicado ao dia quatro e cinco da escala.

O pseudocódigo do procedimento *K-Swap* é apresentado a seguir.

PROCEDIMENTO *K-SWAP*( $k, d$  : inteiros)

**Início**

Inicialize os dados;

**Para**  $j=1$  até  $d-k$  **faça**:

    Construa a matriz de custos  $D$  para o bloco que vai de  $j$  até  $j+k$ ;

    Resolva o PA da matriz  $D$ ;

    Associe os blocos aos enfermeiros vide o resultado obtido;

**Fim.**

O outro procedimento utilizado é denominado PCR ou Procedimento de Cortes e Recombinações. Este método consiste em efetuar cortes entre dois níveis da escala e divide cada uma das  $n$  jornadas em duas jornadas parciais, ficando uma à esquerda e outra à direita do corte. Em seguida é calculada a matriz de custos  $E$ , de dimensões  $n \times n$ , referente às combinações das  $n$  jornadas parciais à esquerda com as  $n$  jornadas parciais a direita do corte. Nessa matriz  $E$ , as jornadas parciais à esquerda são indicadas pelas linhas e as jornadas parciais à direita, pelas colunas. Cada elemento  $e_{ij}$  representa o custo de se associar a jornada parcial à esquerda  $i$  com a jornada parcial à direita  $j$ . Nesse cálculo, o algoritmo verifica quais tarefas fictícias podem ser substituídas para que a recombinação tenha seu custo reduzido. Isso é feito sequencialmente

através dos níveis das jornadas no mesmo sentido em que são feitos os cortes. No valor de  $e_{ij}$  também se incluem penalidades, se houver violações de restrições. Obtida a matriz  $E$ , o PA correlato é resolvido e as jornadas parciais são recombinadas, formando novas jornadas.

Como o algoritmo que resolve o PA entre dois níveis é exato e garante a solução ótima, o custo total da solução diminui ou se mantém a cada corte e recombinação, nunca piora. Uma iteração do PCR consiste em realizar  $d-1$  cortes e recombinções entre os níveis, além de uma recombinação entre cada um dos  $n$  enfermeiros e cada uma das  $n$  jornadas completas. Portanto, uma iteração corresponde a  $d$  cortes e recombinções.

O procedimento PCR é apresentado a seguir.

**PROCEDIMENTO PCR**

**Início**

- Efetue uma divisão na posição de corte 1;
- Gere a matriz de custo  $E$  que relaciona os enfermeiros às jornadas;
- Resolva o PA da matriz  $E$ ;
- Recombine as jornadas aos enfermeiros conforme o resultado obtido;

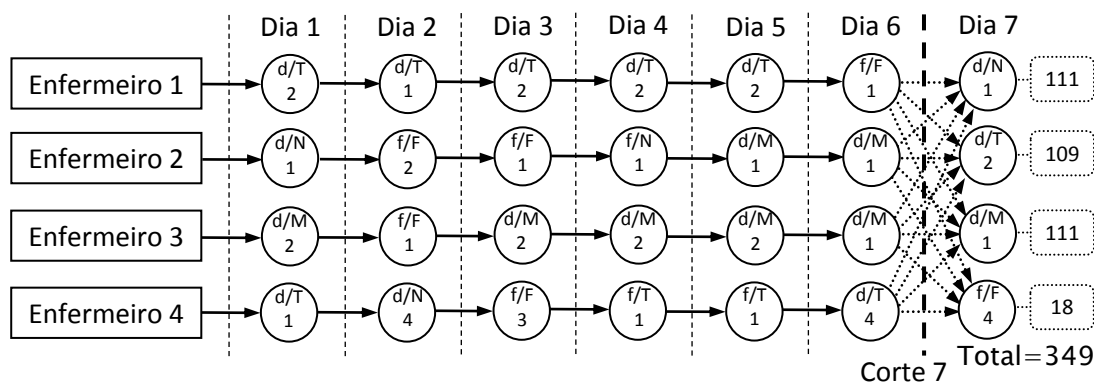
**Para  $l=2$  até  $d$  faça:**

- Efetue uma divisão na posição de corte  $l$ ;
- Gere a matriz de custos  $E$  que relaciona as jornadas parciais;
- Resolva o PA da matriz  $E$ ;
- Recombine as jornadas parciais conforme o resultado obtido;

**Fim Para.**

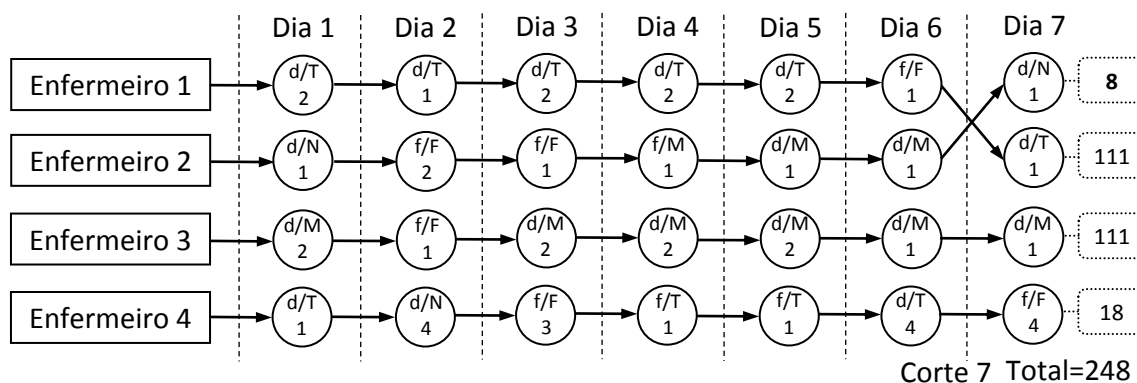
**Fim.**

A Figura 4 ilustra a solução inicial com o corte na escala entre os dias 6 e 7 e as possíveis recombinções. As letras minúsculas  $d$  e  $f$  indicam, respectivamente, se uma tarefa é de demanda ou fictícia. As letras maiúsculas  $M$ ,  $T$ ,  $N$  e  $F$  indicam o tipo de turno, respectivamente, manhã, tarde, noite e folga. O valor imediatamente abaixo desse par de letras explicita o custo daquele turno naquele dia para aquele enfermeiro. À direita constam os custos das jornadas.



**Figura 4: Solução inicial e possíveis recombinções após corte entre os dias 6 e 7.**

A Figura 5 mostra o resultado da aplicação do PCR no corte 7. Com a alteração, a jornada do enfermeiro 2 que, além de uma penalidade, possuía uma soma dos custos de preferência igual a 9, teve uma queda de custo de 109 para 8.



**Figura 5: Recombinação de jornadas parciais pelo PCR utilizando o PA.**

Feita a apresentação dos procedimentos de busca local utilizados, uma visão do procedimento VNS é dada a seguir.

#### PROCEDIMENTO VNS

##### Início

1. Seja  $s_0$  a solução inicial e o conjunto  $r$  de estruturas de vizinhança  $N_k$ , ( $k = 0, 1, \dots, r - 1$ );
2.  $s^* = s_0$ ; {Solução corrente}
3. **Enquanto** (Critério de parada não satisfeito) **faça**:
4.  $k = 0$ ; {Tipo de estrutura de vizinhança}
5. **Enquanto** ( $k \leq r$ ) **faça**:
6. Gere um vizinho qualquer  $s' \in N_k$  a partir de  $s^*$ ;
7.  $s'' = \text{BuscaLocal}(s')$ ; {Algoritmo baseado em VND}
8. **Se** (custo da solução  $s'' <$  custo da solução  $s^*$ ):
9. **Então**  $s^* = s''$ ;  $k = 0$ ;
10. **Senão**  $k = k + 1$ ;
11. **Fim Se**;
12. **Fim Enquanto**;
13. **Fim Enquanto**;
14. **Retorne**  $s^*$ ;

##### Fim.

É na linha 7 que se aplicam os procedimentos de busca local PCR e  $K$ -Swap. Quando o valor da variável  $k$  é 0, o procedimento PCR é executado a partir da solução  $s'$ , obtendo então a solução  $s''$ . Para os demais valores assumidos pela variável  $k$ , o procedimento  $K$ -Swap( $k, d$ ) é executado, onde  $d$  é o número de dias da escala. O procedimento BuscaLocal( $s'$ ) é um algoritmo baseado no método VND (*Variable Neighborhood Descent*), que assim como o VNS, é um método de busca local que explora o espaço de soluções através de trocas sistemáticas de vizinhança. Ele explora a melhor solução de cada vizinhança.

### 3. Análise dos Resultados

Os algoritmos foram implementados utilizando a linguagem de programação Java. Os experimentos foram realizados em dois servidores, um com processador Intel Xeon 10-Core de 2.26 GHz e outro com 2 processadores Intel Xeon 6-Core de 2,93 Ghz, ambos com sistema operacional Windows 2012.

#### 3.1. Instâncias Utilizadas

Foram utilizadas as instâncias fornecidas pela biblioteca digital NSPLib, disponível na página da universidade belga Ugent. Pelo fato de existirem várias versões do problema de escalonamento de enfermeiros, Maenhout e Vanhoucke (2007) propuseram a NSPLib como



forma de padronizar os parâmetros do problema e permitir comparações entre diferentes métodos de resolução. Nela existem arquivos que contêm a demanda para cada dia da escala a ser elaborada e os custos de preferência dos enfermeiros para cada turno de cada dia, com valores inteiros de 1 a 4. Existem, também, arquivos que definem as restrições flexíveis, sendo que cada arquivo constitui um caso de aplicação. No total são 16 casos, sendo os casos de 1 a 8 para escalas de 7 dias e os casos de 9 a 16 para escalas de 28 dias. Os arquivos com demandas e custos permitem a geração de escalas semanais ou quadrissemanais. Para as escalas de 7 dias esses arquivos envolvem 25, 50, 75 ou 100 enfermeiros. Para cada uma dessas quantidades de enfermeiros há 7.290 arquivos diferentes. Cada um desses 29.160 arquivos se relaciona com cada um dos 8 primeiros casos mencionados, possibilitando 233.280 combinações diferentes do problema com escala semanal. Para as escalas de 28 dias existem arquivos que envolvem 30 e 60 enfermeiros, havendo 960 arquivos diferentes para cada quantidade. Esses 1.920 arquivos são associáveis a 8 casos distintos, com numeração de 9 a 16, permitindo 15.360 diferentes problemas com escalas de quatro semanas. Dessa maneira, no total, a NSPLib oferece um universo de 248.640 instâncias diferentes para o problema de escalonamento de enfermeiros.

### 3.2. Resultados e Comparações

Os resultados obtidos pelo algoritmo são comparados com os resultados da base de dados NSPLib e de Constantino *et al.* (2013). As penalidades utilizadas nas soluções alcançadas pelo algoritmo proposto e nos trabalhos citados tiveram o mesmo valor das empregadas na NSPLib, igual a 100. Na biblioteca, esse valor igual a 100 é acrescido ao custo da solução para cada descumprimento de demanda. O algoritmo proposto neste trabalho será identificado como AP-VNS. A Tabela 1 apresenta os valores a serem comparados. Nas linhas estão os resultados relativos a cada uma das quantidades de enfermeiros das instâncias, indicadas na primeira coluna. Na segunda coluna estão as quantidades de dias das escalas. Na terceira, os casos analisados em cada grupo de problemas. Na quarta coluna, o total de instâncias testadas para cada tamanho de problema. Na quinta, sétima e nona colunas constam o custo médio obtido por cada algoritmo na resolução das instâncias daquele grupo. Na sexta, oitava e décima colunas estão as quantidades de soluções factíveis alcançadas.

**Tabela 1: Custos das soluções do AP-VNS, NSPLib e MAPA**

<i>n</i>	<i>d</i>	Caso	Instâncias Testadas	AP-VNS		NSPLib		MAPA	
				Custo Médio	Soluções Factíveis	Custo Médio	Soluções Factíveis	Custo Médio	Soluções Factíveis
25	7	1-8	58,320	256,31	50.320	<b>255,54</b>	<b>50.502</b>	256,65	50.297
50	7	1-8	58.320	<b>507,06</b>	50.795	507,52	<b>50.944</b>	507,62	50.816
75	7	1-8	58.320	<b>771,26</b>	50.519	772,43	<b>50.628</b>	771,67	50.500
100	7	1-8	58.320	<b>1.237,20</b>	51.283	1.239,95	<b>51.350</b>	1.237,95	51.349
30	28	9-16	7.680	<b>1.454,05</b>	<b>5.294</b>	1.492,56	5.167	1.464,75	5.257
60	28	9-16	5.760	<b>2.927,87</b>	4.032	2.995,89	3.989	2.940,40	<b>4.035</b>

A Tabela 1 mostra que, em alguns casos, o método AP-VNS, além de obter um custo menor em relação ao MAPA, ele também obteve uma quantidade maior de soluções factíveis. Para as instâncias de 60 enfermeiros e 28 dias, não foram considerados os casos 13 e 15 por não ter concluído os experimentos até o fechamento deste artigo.

Na Tabela 2 são feitas comparações dos resultados do AP-VNS e os outros dois algoritmos. Esses valores são dados pela Equação 5 a seguir, onde o valor especificado como “OUTRO” faz referência ao valor dos outros algoritmos que devem ser comparados.

$$\text{Gap} = \frac{(\text{Valor AP-VNS}) - (\text{Valor OUTRO})}{\text{Valor OUTRO}} \cdot 100 \quad (5)$$

Para a comparação dos custos médios, as porcentagens positivas indicam o quanto os

custos do AP-VNS ficaram acima dos custos dos outros métodos, enquanto porcentagens negativas indicam o quanto os valores do AP-VNS foram inferiores aos outros métodos. Para o Gap referente à factibilidade das soluções, porcentagens positivas indicam que o número de soluções factíveis encontradas foi maior do que os outros métodos, enquanto porcentagens negativas indicam que o número de soluções factíveis alcançadas foram inferiores em relação aos outros métodos. A Tabela 2 apresenta o *Gap* médio dos algoritmos comparados. A quinta e a sétima colunas apresentam o *Gap* médio da relação dos custos médios obtidos com o algoritmo AP-VNS em relação aos resultados da NSPLib e do método MAPA, respectivamente. A sexta e a oitava colunas mostram o *Gap* médio da relação da quantidade de soluções factíveis obtidas com o algoritmo proposto neste trabalho e os resultados fornecidos pela biblioteca NSPLib e o método MAPA, respectivamente.

**Tabela 2: Relação de custos das soluções e número de soluções factíveis entre os resultados do algoritmo AP-VNS em relação aos resultados da NSPLib e MAPA**

<i>n</i>	<i>d</i>	Caso	Instâncias Testadas	AP-VNS/NSPLib (%)		AP-VNS/MAPA(%)	
				<i>Gap</i> Custo Médio	<i>Gap</i> Soluções Factíveis	<i>Gap</i> Custo Médio	<i>Gap</i> Soluções Factíveis
25	7	1-8	58.320	0,28	-0,39	<b>-0,14</b>	<b>0,06</b>
50	7	1-8	58.320	<b>-0,10</b>	-0,32	<b>-0,11</b>	-0,04
75	7	1-8	58.320	<b>-0,16</b>	-0,23	<b>-0,06</b>	<b>0,05</b>
100	7	1-8	58.320	<b>-0,23</b>	-0,15	<b>-0,06</b>	-0,13
30	28	9-16	7.680	<b>-2,54</b>	<b>2,57</b>	<b>-0,73</b>	<b>0,71</b>
60	28	9,16*	5.760	<b>-2,27</b>	<b>1,08</b>	<b>-0,43</b>	-0,07

\*Exceto os casos 13 e 15.

É possível observar que o algoritmo proposto AP-VNS obteve custos menores em relação aos resultados da NSPLib, no entanto, alcançou um número inferior de soluções factíveis. Em relação ao método MAPA, os resultados apresentam uma redução no custo médio das soluções e em alguns casos um número maior de soluções foi alcançado. Para o caso complexo de 60 enfermeiros em 28 dias, o algoritmo mostrou um desempenho superior em relação aos outros algoritmos. Nota-se uma grande melhora tanto na redução dos custos, quanto no número de soluções factíveis alcançadas. Foi possível observar que o algoritmo proposto consegue um desempenho melhor conforme a dimensão do problema aumenta.

#### 4. Conclusões

O presente trabalho teve como objetivo investigar a aplicação da meta-heurística VNS na fase de melhoria da solução para o problema de escalonamento de enfermeiros com o objetivo de minimizar o custo das soluções. O algoritmo proposto AP-VNS combina a meta-heurística VNS com os procedimentos de busca local denominados *K-Swap* e PCR. Essa nova proposta foi experimentada sobre uma base de dados conhecida na literatura e que foi criada com o intuito de padronizar os parâmetros do problema e permitir que os diferentes métodos de resolução existentes possam ser comparados. A partir da realização dos experimentos com o algoritmo proposto (AP-VNS) foi possível constatar uma melhora na qualidade das soluções quando comparado com o algoritmo MAPA, e uma melhora também em alguns casos quando comparados com os resultados apresentados pela biblioteca NSPLib. O algoritmo mostrou um desempenho superior quando aplicado nas instâncias de grande porte do problema.

Como trabalhos futuros pretende-se ampliar os experimentos desse algoritmo com instâncias maiores e de outras bases de dados de benchmark.

#### 5. Agradecimentos

Os autores agradecem a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro.

## Referências

- Aickelin, U. e Downsland, L.** (2004), An indirect algorithm for a nurse-scheduling problem. *Computers & Operations Research*, Elsevier, 31, 761-778.
- Awadallah, M., Khader, A., Al-Betar, M. e Bolaji, A.** (2011), Nurse scheduling using harmony search. *Bio-Inspired Computing: Theories and Applications (BIC-TA)*, 6, 58-63.
- Bellanti, F., Carello, G., Della Groce, F. e Tadei, R.** (2004), A greedy-based neighborhood search approach to a nurse rostering problem. *European Journal of Operational Research*, 153, 28-40
- Burke, E. K., Causmaecker, P., Berghe, G. V. e Landeghem, H.** (2004a), The state of the art of nurse rostering. *Journal of Scheduling*, 7(6), 441-499.
- Burke, E., De Causmaecker, P. Petrovic, S. e Berghe, G.** (2004b), Variable neighborhood search for nurse rostering problems, *Metaheuristics: Computer Decision-Making*, 153-172
- Burke E. K., Cururtois, T., Post, G., Qu, R. e Veltman, B.** (2008), A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem, *European Journal of Operations Research*, 188(2), 330-341.
- Carpaneto, G. e Toth, P.** (1987), Primal-dual algorithms for the assignment problem, *Discrete Applied Mathematics*, 18, 137-153.
- Constantino, A. A., Landa-Silva, D., Melo, E. L., Mendonça, C. F. X., Rizzato, D. B. e Romão, W.** (2013), A heuristic algorithm based on multi-assignment procedures for nurse scheduling, *Annals of Operations Research*, 205, 1-19.
- Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B. e Sier, D.** (2004), An annotated bibliography of personnel scheduling and rostering, *Annals on Operations Research*, 127, 21-144.
- Garey, M. R. e Johnson, D. S.**, *Computers and intractability: a guide to the theory of NP-Completeness*, W. H. Freeman, New York, 1979.
- Gutjahr, W. e Rauner, M.** (2007), An ACO algorithm for a dynamics regional nurse-scheduling problem in Austria. *Computers & Operations Research*, Elsevier, 34, 642-666.
- Kuhn, H. W.** (2005), "The Hungarian method for the assignment problem", *Naval Research Logistics*, 52(1):7-21. [doi:10.1002/nav.20053](https://doi.org/10.1002/nav.20053)
- Maenhout, B. e Vanhoucke, M.** (2007), An electromagnetic meta-heuristic for the nurse scheduling problem, *Journal of Heuristics*, 13(4), 359-385.
- Maenhout, B. e Vanhoucke, M.** (2008) Comparison and hybridization of crossover operators for the nurse scheduling problem, *Annals of Operations Research*, 159(1), 333-353.
- Maenhout, B. e Vanhoucke, M.** (2006), New Computational Results for the Nurse Scheduling Problem: A Scatter Search Algorithm, *Lecture Notes in Computer Science*, 3906, 159-170.
- Nedas, K.** (2005) Implementation of Munkres-Kuhn (Hungarian) algorithm. Disponível em: <http://konstantinosnedas.com/dev/soft/munkres.htm>, acesso em 26/08/2013.
- Ohki, M., Morimoto, A. e Miyake, K.** (2006), Nurse scheduling by using cooperative GA with efficient mutation and mountain-climbing operators intelligent systems. *3<sup>rd</sup> International IEEE Conference*, 3, 2006. 164-169.
- Osogami, T. e Imai, H.** (2000), Classification of Various Neighborhood Operations for the Nurse Scheduling Problem, *Lecture Notes in Computer Science*, 1969, 72-83.
- Tsai, C. e Li, S.** (2009), A two-stage modeling with genetic algorithms for the nurse scheduling problem, *Expert Systems with Applications*, 36(5), 9506-9512.