

---

## **APLICANDO BUSCA TABU E ALGORITMO GENÉTICO NO CORPO DE ENFERMAGEM DE UM HOSPITAL UNIVERSITÁRIO**

---

CELSO CAMILO, DIOGO STELLE, DIEGO ROZA

*Resumo: o escalonamento do corpo de enfermagem é uma tarefa custosa e complexa, por isso causa uma perda de tempo e atenção de um funcionário importante do hospital, normalmente enfermeiro-chefe. Por isso, este trabalho propõe a aplicação do Algoritmo Genético (AG) e da Busca Tabu (BT) para solucionar o problema de escalonamento do corpo de enfermagem de um Hospital Universitário. Os resultados mostram a superioridade do método BT para resolução do problema abordado.*

*Palavras-chave: Algoritmo Genético, Busca Tabu, Programação de Horários, Enfermagem, Metaheurística*

**N**a literatura é comum encontrar trabalhos que tratam problemas da classe *Timetabling* por meio das técnicas AG e BT, entre eles destacam-se: escalonamento de horários (MARIANO, 2007; NETO, CONSTANTINO, ARAUJO, 2006; FREITAS *et al.*, 2007), escalonamento de salas de aulas (SOUZA; MARTINS; ARAÚJO, 2002), escalonamento de motoristas (MARINHO *et al.*, 2004), escalonamento de enfermeiros (SANTOS, 2006; DOWSLAND, 1998), entre outros.

Dentre estes, o problema de escalonamento de trabalho para profissionais de enfermagem, conhecido na literatura como *Nurse Rostering Problem* ou *Nurse Scheduling Problem*, trata da elaboração de escalas de trabalho para profissionais de enfermagem de hospitais. Por se tratar de um problema muito custoso e restritivo é objeto de estudo de muitos pesquisadores (RANGEL, ÉVORA, 2007; DOWSLAND, 1998; IKEGAMI, NIWA, 2003; MILLAR, KIRAGU, 1998) da área de pesquisa operacional.

Nas unidades hospitalares, geralmente, a formulação da escala de trabalho dos profissionais é elaborada manualmente por um profissional que tenha conhecimento do hospital e de seus profissionais. Essa atividade demanda muito tempo de quem a executa, pois existem diversas restrições e preferências a serem atendidas, e muitas vezes a alteração de algum desses valores pode ocasionar conflitos com a escala elaborada até o momento.

Por isso e dando continuidade ao trabalho Camilo e Stelle (2008), este trabalho tem como objetivo comparar os resultados obtidos com a aplicação das metaheurísticas Algoritmos Genéticos (AG) e Busca Tabu (BT) ao problema de elaboração de escalas de trabalho para os profissionais de enfermagem do HU-Dourados.

Para avaliar os algoritmos elaborou-se dois cenários distintos, sendo um baseado no cenário real do HU-Dourados e outro com modificações na quantidade mínima e máxima de profissionais por setor/turno de trabalho. Na seção 6 encontra-se a descrição de ambos os cenários.

Após a realização dos testes e análise do desempenho dos algoritmos, pode-se observar que o BT apresentou-se, nos cenários estudados, como a melhor alternativa para solucionar o problema de elaboração de escalas de trabalho para profissionais de enfermagem do HU-Dourados.

Este trabalho está organizado da seguinte forma. Na seção 2 o conceito de algoritmos genéticos (BRUN, 2004; GROSKO; GORSKI; DIAS, 2006) é abordado; na seção 3 descreve-se a organização e funcionamento do BT; na seção 4 apresenta-se definições do problema abordado; na seção 5 é descrito o problema do HU-Dourados; a seção 6 mostra os resultados da simulação da ferramenta; e, por fim, as conclusões são expostas na seção 7.

## ALGORITMOS GENÉTICOS

Algoritmo Genético (AG) é um método de busca e otimização inspirado na teoria da evolução natural das espécies de Darwin (BRUN, 2004). O AG trabalha com uma população de indivíduos, onde cada indivíduo representa uma possível solução para o problema de otimização que esta sendo tratado. A figura 1 apresenta a estrutura básica de um AG, já as fases do fluxograma serão descritas nas próximas seções.

### Representação

A escolha da representação do indivíduo (cromossomo) é um passo fundamental na implementação do AG, pois influencia no espaço de busca a ser varrido pelo algoritmo e na codificação dos operadores de *crossover* e mutação que serão utilizados (BARRA, 2007). Os indivíduos podem ser representados na forma de cadeia de bits, caracteres ou números, que representam os genes biológicos. A figura 2 representa um indivíduo codificado em forma de cadeia de bits.

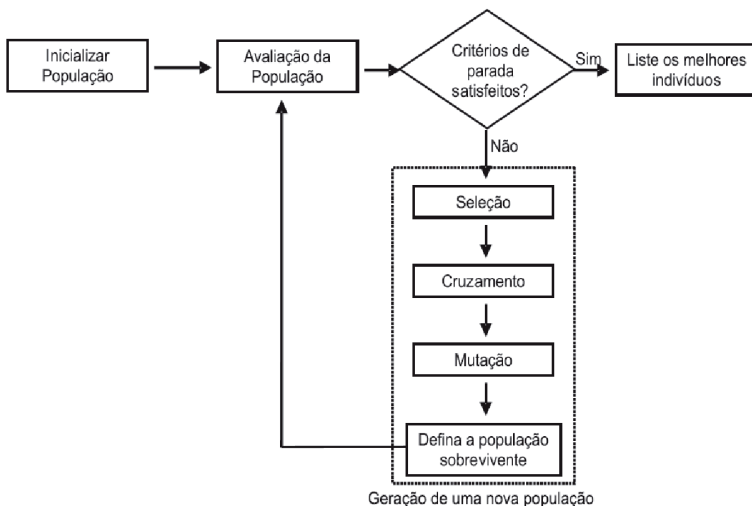


Figura 1: Fluxo de execução do AG

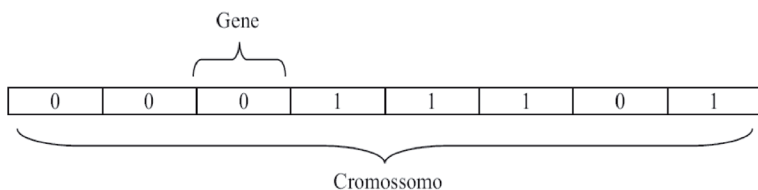


Figura 2: Indivíduo representado na forma de cadeia de bits

## Inicializar População

Após definir como representar os indivíduos, o próximo passo é a criação da população inicial de indivíduos, que serão pontos de partida na busca por melhores soluções para o problema de otimização a ser tratado.

Geralmente esta população é criada aleatoriamente, porém existem outros métodos para gerar esta população, entre eles: o uso de heurísticas e a criação de  $n/2$  indivíduos aleatoriamente, seguido da inversão dos bits destes para formar os indivíduos restantes para completar a população.

## Avaliação da População

Cada indivíduo da população é avaliado segundo uma função objetiva (FO) do problema de otimização a ser tratado, esta função determina quão bom é o indivíduo para solucionar o problema. Esse cálculo atribui ao indivíduo um valor correspondente a seu benefício em relação à FO, esse valor é denominado aptidão.

## Seleção

O operador de seleção tem como objetivo selecionar, dentre toda a população, indivíduos para gerarem descendentes durante o processo de evolução. Existem diversos métodos que podem ser utilizados pelo AG para realizar este processo, dentre eles: seleção por ranking, seleção Bi-classista, seleção por diversidade, seleção local, seleção truncada e a seleção por torneio, porém o mais utilizado é o método de seleção por roleta (PAPPA, 2002; MARIANO, 2007; LINDEN, 2006).

No método de seleção por roleta, cada indivíduo da população ocupa um espaço na roleta proporcional ao valor da sua aptidão, sendo assim os indivíduos mais aptos possuem mais chances de serem selecionados.

Pode-se definir a probabilidade de seleção  $p_i$ , de um indivíduo  $i$ , com aptidão  $f(x_i)$ , onde  $k$  representa o total de indivíduos como:

$$p_i = \frac{f(x_i)}{\sum_{k=1}^N f(x_k)} \quad (1)$$

Nessa etapa ocorre a recombinação das características genéticas de dois indivíduos (pais selecionados). Existem várias técnicas que podem ser utilizadas para realizar essa recombinação, dentre elas destacam-se: *crossover* de dois pontos, uniforme, baseado em maioria, de ordem e de ciclo, no entanto, o método mais utilizado para a codificação binária é o cruzamento de ponto único (MARIANO, 2007; LINDEN, 2006).

No cruzamento de ponto único, *crossover* clássico, é escolhido aleatoriamente um ponto de corte, o qual divide um indivíduo em duas partes, junta-se a parte anterior a esse ponto de um indivíduo pai com a parte posterior desse ponto do outro indivíduo pai, gerando assim, dois novos indivíduos formados pelas características genéticas de ambos os indivíduos progenitores. O cruzamento tenta convergir os indivíduos para um ponto de busca desejável.

Um exemplo do operador de *crossover* clássico para codificação binária pode ser visto na figura 3

P1 = (1010|0001)

P2 = (1000|1101)



F1 = (1010|1101)

F2 = (1000|0001)

Figura 3: Operador de cruzamento clássico para codificação binária

## Mutação

Nessa etapa altera-se aleatoriamente o valor de uma ou mais posições (genes) de um indivíduo. Esse processo tem como objetivo induzir o algoritmo a explorar locais de busca até então desconhecidos pelo algoritmo, em busca de melhores soluções para o problema. A figura 4 demonstra um exemplo do operador de mutação clássico para codificação binária.

$$\begin{array}{c} P1 = (1000|1101) \\ \downarrow \\ P1 = (1011|1101) \end{array}$$

Figura 4: Operador de mutação clássico para codificação binária

## Definição da População Sobrevivente

Nessa etapa escolhem-se os indivíduos que farão parte da nova população. Na forma mais simples desse procedimento é apenas feita a troca de todos os indivíduos da geração anterior pelos novos indivíduos criados. Essa substituição também pode ser realizada por meio de procedimentos bastante conhecidos na literatura como elitismo e *steady-state*.

O Elitismo é usado para garantir que o desempenho do AG ao longo do processo evolutivo sempre cresça. Preservam-se os  $n$  melhores indivíduos de cada geração ( $g$ ), transferindo-os para a próxima geração ( $g+1$ ).

Encontra-se na literatura distintas definições sobre a abordagem *steady-state*, são elas:

- São criados  $n$  indivíduos através dos operadores de crossover e mutação, com  $n <$  tamanho da população. Em seguida, os mesmos são avaliados, e se os indivíduos criados forem mais aptos que os piores indivíduos da geração anterior, substituem-se pelos criados. Nesse método, preserva-se praticamente toda a população anterior (SHIZEN; ZHOUYANG, 2004; MOGNON, 2004);
- Permite a interação entre indivíduos de duas gerações distintas. Essa abordagem permite a reprodução entre indivíduos recém

criados e outros da geração anterior. Cria-se os filhos (geralmente de 2 em dois) e substituem os piores pais por estes novos indivíduos (LINDEN, 2006).

### Parâmetros do AG

Ao implementar o AG deve-se considerar alguns parâmetros (BRUN, 2004) que influenciarão diretamente o funcionamento do mesmo. São eles:

- **Tamanho da população:** influência diretamente na exploração do espaço de busca, no tempo de execução e na demanda por recursos computacionais. Caso defina-se um valor muito baixo para esse parâmetro, limita-se o espaço de busca por possíveis soluções do problema, porém se este valor for muito alto, implicará na necessidade de mais recursos computacionais ou no aumento do tempo necessário para execução do algoritmo.
- **Taxa de cruzamento:** define com qual probabilidade ocorrerá o cruzamento entre os indivíduos. Quanto maior for esta taxa, mais rapidamente serão introduzidos novos indivíduos na população, porém se a mesma for muito alta há possibilidade de ocorrer perda de indivíduos com alto grau de aptidão prematuramente.
- **Taxa de mutação:** define a probabilidade de um indivíduo ter um ou mais de seus genes alterados pelo operador de mutação. Sua função é prevenir que os indivíduos se aglomerem em ótimos locais, induzindo-os a explorarem pontos no espaço de busca das possíveis soluções do problema, até então desconhecidos. A escolha de um valor muito alto para essa taxa induzirá o AG a uma busca aleatória.
- **Número de gerações:** é um dos critérios de parada do AG, representa o número de ciclos que o algoritmo será executado. Influencia diretamente no espaço de busca por possíveis soluções, no tempo de execução e na demanda por recursos computacionais. Quanto maior for o valor definido para este parâmetro maior será a exploração do espaço de busca por possíveis soluções para o problema e conseqüentemente maior será a demanda por recursos computacionais ou maior será o tempo necessário para a execução do AG.
- **Intervalo de geração:** determina a percentagem de indivíduos

que serão substituídos na troca de uma geração por outra. Há diversas maneiras de realizar esta troca, entre elas: substituição total, com elitismo, *steady-state*, entre outras.

## Busca Tabu

A busca tabu foi descrita inicialmente por Glover (GLOVER, 1990) em 1986, como uma técnica de busca local, com grande capacidade de fugir do ótimos locais e atingir ótimos globais. Utiliza-se de uma estrutura de memória, para guardar atributos das soluções já encontradas, evitando assim que o método entre em ciclagem (retorne a soluções já visitadas); e aceita movimentos de piora para poder explorar novas regiões (GLOVER; LAGUNA, 1993).

As classes de problemas tratados pelo BT tem, normalmente, dois tipos de restrições: *Hard Constraints* (HC) e *Soft Constraints* (SC), sendo a HC restrições obrigatórias, que devem ser atendidas para que a solução seja factível, e a SC restrições desejáveis, que podem ser violadas. Entretanto, quanto mais SC forem atendidas melhor se tornará à solução.

## Vizinhança

A exploração do espaço de busca é feita através da vizinhança, estabelecida por um raio. A partir de uma solução  $S$ , aplica-se um movimento  $M(S)$  que resulta em  $S''$ . A definição do movimento está diretamente ligada à representação da solução. Duas das técnicas de movimento utilizadas, principalmente em problemas de programação de horários, são troca e a realocação de valores. Essa movimentação deve ser capaz de explorar todo o espaço amostral. Para ilustrar os movimentos, foi utilizado o problema de alocação de salas, onde a solução do problema é representada por uma matriz de Horários x Sala e as turmas são alocadas ou realocadas pelos movimentos, como observa-se na figura 5.

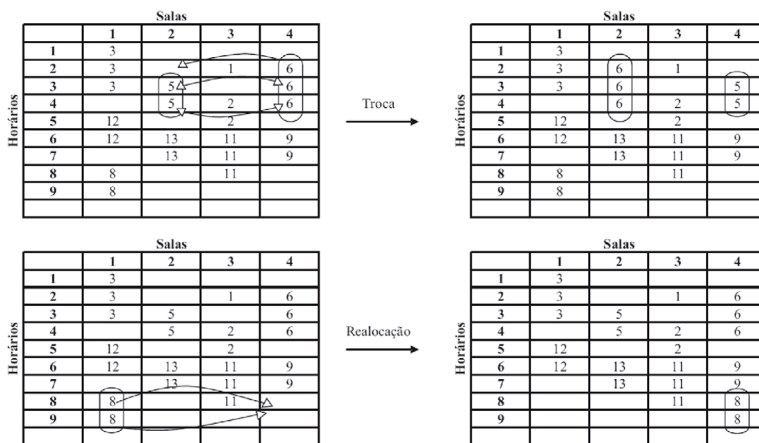


Figura 5: Movimentos de troca e realocação

### Lista Tabu

A principal característica da BT é a Lista Tabu (LT), local onde armazena-se as informações das soluções já percorridas. Com esses dados, cria-se uma lista de movimentos tabu's, assim evita-se que o algoritmo retorne a soluções já visitadas. Essa lista tem tamanho pré-definido ou pode ser criada dinamicamente, pois se torna inviável armazenar os dados de todas as soluções percorridas.

O objetivo da LT é não permitir que o inverso de um movimento aconteça, assim ela evita que o algoritmo fique preso em um ótimo local. Há casos em que um movimento mesmo sendo tabu é escolhido. Isso ocorre quando o valor dessa solução é maior que a melhor solução encontrada até o presente momento, esse procedimento é conhecido como critério de aspiração. O tempo que um movimento se torna tabu esta diretamente relacionada ao tamanho da LT, assim um movimento, em uma lista de tamanho T, será tabu por T iterações.

### Funcionamento do Algoritmo

A estrutura do BT é ilustrada na figura 6, nela identifica-se os seguintes passos:

1. Cria-se a solução inicial, calcula-se sua Função Objetivo (FO) que passa a ser a Melhor Solução (MS), e a Solução Corente (SC);
2. Cria-se a vizinhança com seus respectivos valores da FO. Dessas, é escolhida a solução com maior valor de FO e que não esteja na LT. Caso um desses vizinhos seja maior que a MS, mesmo estando na LT, ele será escolhido (condição de aspiração);
3. O escolhido passará a ser, a SC e caso seja a melhor solução encontrada até agora, atualiza-se a MS, e a variável que informa em qual iteração a MS foi encontrada;
4. Atualiza-se a LT;
5. Caso a expressão “contador de Iteração - Iteração que encontrou a MS > BTMax” seja falsa, retorna para o item 1; caso positiva, finaliza a execução e retorna a MS.

**procedimento BT**

1. Seja  $s_0$  solução inicial;
  2.  $s^* \leftarrow s$ ;     {Melhor solução obtida até então}
  3.  $\text{Iter} \leftarrow 0$ ;    {Contador do número de iterações}
  4.  $\text{MelhorIter} \leftarrow 0$ ;   {Iteração mais recente que forneceu  $s^*$ }
  5. Seja  $B \text{ Tmax}$  o número máximo de iterações sem melhora em  $s^*$ ;
  6.  $T \leftarrow \emptyset$ ;     {Lista Tabu}
  7. Inicialize a função de aspiração  $A$ ;
  8. enquanto ( $\text{Iter} - \text{MelhorIter} \leq B \text{ Tmax}$ ) faça
  9.      $\text{Iter} \leftarrow \text{Iter} + 1$ ;
  10.    Seja  $s' \leftarrow s \oplus m$  o melhor elemento de  $V \subseteq N(s)$  tal que o movimento  $m$  não seja tabu ( $m \notin T$ )  
       ou  $s'$  atenda a condição de aspiração ( $f(s') < A(f(s))$ );
  11.    Atualize a Lista Tabu  $T$ ;
  12.     $s \leftarrow s'$ ;
  13.    se  $f(s) < f(s^*)$  então
  14.        $s^* \leftarrow s$ ;
  15.        $\text{MelhorIter} \leftarrow \text{Iter}$ ;
  16.    fim-se;
  17.    Atualize a função de aspiração  $A$ ;
  18. fim - enquanto;
  19. Retorne  $s^*$ ;
- fim BT**;

Figura 6: Algoritmo padrão da BT

## O PROBLEMA DO ESCALONAMENTO DE ENFERMEIROS

É comum encontrar trabalhos da área de Pesquisa Operacional relacionados à resolução de problemas como: escalonamento de horários (MARIANO, 2007; NETO, CONSTANTINO; ARAUJO, 2006; FREITAS *et al.*, 2007), escalonamento de salas de aulas (SOUZA, MARTINS, ARAÚJO, 2002), escalonamento de motoristas (MARINHO *et al.*, 2004), escalonamento de enfermeiros (SANTOS, 2006; DOWS- LAND, 1998), entre outros.

O problema de escalonamento de enfermeiros é complexo, pois normalmente atende-se a diversas restrições legais e de exigências dos profissionais como: preferências por turno e setor de trabalho, a quantidade mínima e máxima de profissionais permitida por turno e setor de trabalho, entre outras.

O enfermeiro-chefe, geralmente o responsável por elaborar as escalas de trabalhos dos profissionais de enfermagem de uma unidade hospitalar, se depara rotineiramente com a difícil tarefa de elaborar escalas de trabalhos que atendam a demanda de profissionais nos setores e turnos de trabalho e as restrições impostas pelo Conselho de Enfermagem e internas da unidade hospitalar, além de tentar atender ao máximo as preferências dos profissionais.

Visando facilitar essa tarefa, esse trabalho tem como base propor e analisar os resultados obtidos com a geração das escalas de trabalhos do HU-Dourados por meio de duas ferramentas computacionais: AG e BT.

## O PROBLEMA DE ESCALONAMENTO DE ENFERMEIROS DO HOSPITAL UNIVERSITÁRIO DE DOURADOS

O Hu-Dourados funciona 24 horas por dia e atende a população da região da Grande Dourados, formado por 12 municípios e 295.338 pessoas (CIDADANIA, 2008).

O corpo de enfermagem do hospital conta atualmente com 30 enfermeiros, os quais trabalham seguindo as regras impostas pelo Conselho Federal de Enfermagem e demais regras internas do hospital, e com 122 técnicos de enfermagem, os quais também seguem estas regras, mas fazem uma escala mais ajustada pela demanda do setor locado e o fluxo de pacientes no mesmo.

Há quatro turnos distintos de trabalho no hospital são eles: manhã, tarde e noite, porém as noites são subdivididas em noites pares e ímpares. Esta subdivisão facilita o controle de uma regra trabalhista, de que um profissional que trabalhou em turno de noite ímpar não possa trabalhar em um turno de noite par.

O HU-Dourados conta com uma enfermaria dividida em 6 setores: Posto IV e III da Clínica Médica, Posto II da Clínica Cirúrgica, Posto I da Clínica Pediátrica, Unidade de Terapia Intensiva Pediátrica e Centro Cirúrgico. Em cada setor existem profissionais responsáveis pelo mesmo (enfermeiro-coordenador).

Como as restrições não são iguais para todos os meses, cada fim de mês a enfermeira-chefe tem o dever de elaborar as escalas de trabalho para o mês seguinte e se depara novamente com uma série de necessidades que terá que atender. Mesmo com a experiência obtida pela profissional com a elaboração de escalas anteriores, essa tarefa ainda demanda muito tempo para ser executada, pois para que a nova escala seja considerada viável ela precisa atender uma série de restrições HC e SC.

As restrições *hard* e *soft constraints* encontradas no HU-Dourados foram:

– *Hard Constraints*

- Deve-se respeitar o número mínimo de profissionais necessários para um bom funcionamento do setor/turno, também garantindo que o número máximo não seja ultrapassado;
- Verificar a indisponibilidade do profissional em trabalhar em determinado setor/turno, o que pode ocorrer, por exemplo, no caso de necessidade de formação especializada ou incompatibilidade de perfil;
- Não locar funcionários em férias, licença ou outros;
- Deve-se garantir que o profissional seja escalonado em apenas um turno por dia, já que a carga horária é de seis horas diárias.

– *Soft Constraints*

- Deve-se tentar atender ao máximo as preferências dos funcionários, definida em relação ao setor/turno de trabalho;
- Deve-se tentar alocar o maior número de especialistas nos respectivos setores das especialidades;

- Escalonar os profissionais de modo que o número de funcionários em um setor/turno fique próximo da média entre o mínimo e o máximo de profissionais permitidos naquele setor/turno. Garantindo assim, uma distribuição mais homogênea dos funcionários entre as unidades.

## TESTES E RESULTADOS

As soluções obtidas são avaliadas segundo a função objetivo  $FO(s)$  (equação 2), onde  $s$  representa uma solução:

$$FO(s) = F1(s) + F2(s) \quad (2)$$

O objetivo da função  $F1(s)$ , ver equação 3, é tentar aproximar a quantidade total de profissionais alocados em um setor/turno (ST) da média, calculada pela quantidade mínima e máxima permitida no mesmo. Para isso, penaliza-se o setor/turno onde a quantidade de profissionais alocados está distante da média, visando equilibrar a quantidade de profissionais alocados nos STs.

$$F1(s) = \sum_{ST=1}^m F3(ST) \quad (2)$$

sendo  $m$  a quantidade de STs.

A função  $F3(ST)$ , equação 4, retorna o valor de penalização aplicado a um determinado setor/turno (ST) que não contribuiu para o balanceamento da distribuição dos funcionários.

$$F3(ST) = |(((stmin + stmax)/2) - qtdFuncST) - pena| \quad (4)$$

sendo  $pena = 1$ ,  $stmin$  e  $stmax$  quantidade mínima e máxima de profissionais permitidos em um setor/turno (ST) e  $qtdFuncST$  a quantidade total de profissionais alocados em um ST.

Define-se  $F2(s)$ , como sendo a soma dos valores  $g(s)_k$ , para cada profissional (k), sendo  $g(s)_k$  o resultado da multiplicação da preferência ( $preffunc[k,ST]$ ) do profissional (k) no setor/turno (ST) por um peso ( $pesost$ ) pré-estabelecido, ver equação 5.

$$F2(s) = \sum_{k=1}^n g(s)_k \quad (5)$$

sendo  $\text{pesost}=10$ ,  $g(s)_k = (\text{preffunc}[k,ST] * \text{pesost})$  e  $n =$  quantidade total de profissionais.

A função  $F2(s)$  visa maximizar as preferências por ST atendidas dos profissionais na solução. Quanto maior for a quantidade de preferências atendidas maior será o valor de retorno da função.

Os parâmetros “pena”, na  $F3$  que compõe a  $F1$ , e “pesost” determinam o peso que as funções  $F1$  e  $F2$  tem na função objetivo  $FO(s)$ . Com isso, é possível privilegiar soluções que atendam mais as preferências dos funcionários ( $F2$ ) ou o balanço da distribuição ( $F1$ ).

Os testes realizados com os algoritmos AGs e BT utilizaram como base de informações dois cenários distintos, descritos abaixo.

No cenário 1, cenário real do HU-Dourados, considera-se para os testes 122 profissionais que precisam ser escalonados em 4 turnos e em 6 setores de trabalho distintos. Como na atual situação do hospital há um número exato de profissionais necessários para suprir as demandas dos mesmos, nos diversos setores e turnos, as restrições de quantidade mínima e máxima de profissionais por setor/turno são iguais. Ou seja, a soma dos mínimos é igual ao total de profissionais. O ótimo global neste cenário é atingido com  $FO(s) = 4880$ .

Foram realizados 3 testes com cada algoritmo, sendo que, cada teste constitui-se da execução do algoritmo por 10 vezes. A diferença entre os testes está no valor dos parâmetros dos algoritmos. Os resultados obtidos com os métodos AG e BT estão representados respectivamente nas tabelas 1 e 2.

Tabela 1: Configurações do Cenário 1 para o AG

Cenário 1					
PARAMÊTROS DO AG					
Teste 1	Tamanho da População	Probabilidade de <i>crossover</i>	Probabilidade de mutação	Condição de parada	Elitismo
	100	70/100	1/1000	500 gerações	
Teste 2	Tamanho da População	Probabilidade de <i>crossover</i>	Probabilidade de mutação	Condição de parada	Elitismo
	100	70/100	1/100	500 gerações	
Teste 3	Tamanho da População	Probabilidade de <i>crossover</i>	Probabilidade de mutação	Condição de parada	Elitismo
	100	70/100	5/100	500 gerações	
	Menor <i>FO(s)</i>	Maior <i>FO(s)</i>	Média <i>FO(s)</i>		
	2700	2820	2762		1
	2700	2860	2764		1
	2720	2880	2794		1

Tabela 2: Configurações do Cenário 1 para o BT

Cenário 1				
PARAMÊTROS DA BT				Média iteração
Teste 1	Lista Tabu	Vizinhança	Condição de parada	para alcançar ótimo global
	5	50	500 iterações	
Teste 2	Lista Tabu	Vizinhança	Condição de parada	para alcançar ótimo global
	10	50	500 iterações	
Teste 3	Lista Tabu	Vizinhança	Condição de parada	para alcançar ótimo global
	5	120	500 iterações	
	Menor <i>FO(s)</i>	Maior <i>FO(s)</i>	Média <i>FO(s)</i>	
	4620	4780	4706	307
	4540	4820	4686	338
	4780	4880	4836	500

A figura 7 representa a comparação dos resultados obtidos pelas técnicas AG e BT em relação ao teste, e respectiva execução, que obteve a melhor *FO(s)*, no caso do AG teste 3 execução 1 e no caso do BT teste 3 execução 7. A figura 8 também ilustra a comparação, entre os algoritmos, no melhor teste de cada, no entanto, neste caso segundo as médias das execuções por iteração.

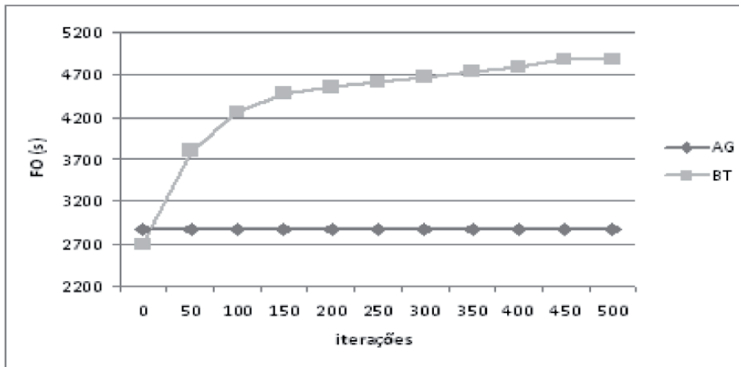


Figura 7: Comparação do desempenho das técnicas AG e BT no Cenário 1, segundo a melhor execução de cada

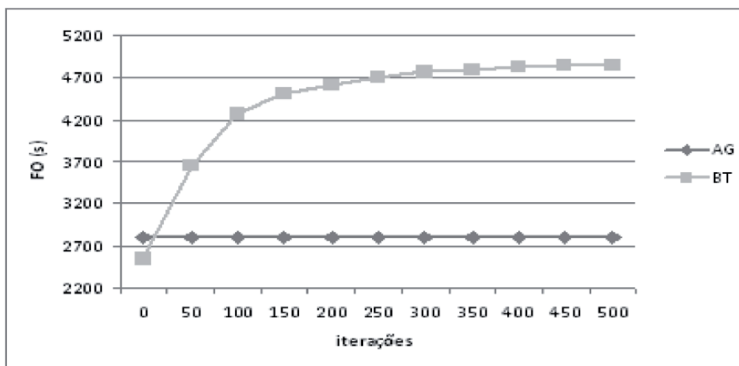


Figura 8: Comparação do desempenho das técnicas AG e BT no Cenário 1 teste 3, segundo a média das execuções do melhor teste de cada

Visualiza-se facilmente por meio da figura 7 que o AG ao longo das iterações não conseguiu melhorar a solução inicial encontrada, este fato pode ser justificado pelas restrições de quantidade mínima de profissionais necessários para trabalhar em um ST e pelas restrições de preferências dos profissionais por ST, combinadas com o operador de *crossover* utilizado, que faz a troca de blocos de genes dos pais, o que ocasiona, na maioria dos casos, a geração de novas soluções infactíveis.

Como na situação real do hospital o número de profissionais disponíveis para trabalhar em um ST é igual a demanda do ST

( $stmin = stmax$ ), a operação de combinar as soluções geradas pelo AG ocasionam novas soluções que não respeitam a quantidade mínima de profissionais exigida em um ST ou violam as restrições de preferências dos funcionários por ST, alocando-os em ST cujo mesmo tem indisponibilidade para trabalhar.

Já o procedimento BT, teve ótima performance na resolução do problema. Como pode-se observar na figura 7, a  $FO(s)$  inicia com valor de 2.700 e ao final atinge 4.880, o ótimo global.

No cenário 2, elaborado especialmente para os testes, modificou-se apenas a quantidade mínima e máxima de profissionais por setor/turno (ST). Aumentou-se a faixa de valores entre a quantidade mínima e a máxima, possibilitando assim, um maior espaço de busca para o AG e uma condição mais favorável para o operador de *crossover* utilizado. As tabelas 3 e 4 representam, respectivamente, os resultados obtidos pelo AG e BT no cenário 2.

Tabela 3: Configurações e resultados do Cenário 2 para o AG

Cenário 2					
PARAMÊTROS DO AG					
<i>Teste 1</i>	Tamanho da População 100 Menor $FO(s)$ 3372	Probabilidade de <i>crossover</i> 70/100 Maior $FO(s)$ 3540	Probabilidade de mutação 1/1000 Média $FO(s)$ 3475	Condição de parada 500 gerações	Elitismo 1
<i>Teste 2</i>	Tamanho da População 100 Menor $FO(s)$ 3816	Probabilidade de <i>crossover</i> 70/100 Maior $FO(s)$ 4098	Probabilidade de mutação 1/100 Média $FO(s)$ 3959	Condição de parada 500 gerações	Elitismo 1
<i>Teste 3</i>	Tamanho da População 100 Menor $FO(s)$ 2900	Probabilidade de <i>crossover</i> 70/100 Maior $FO(s)$ 3344	Probabilidade de mutação 5/100 Média $FO(s)$ 3112.2	Condição de parada 500 gerações	Elitismo 1

Tabela 4: Configurações e resultados do Cenário 2 para o BT

Cenário 2				
PARAMÊTROS DA BT				
Teste 1	Lista Tabu	Vizinhança	Condição de parada	para alcançar ótimo global
	5 Menor $FO(s)$ 4602	50 Maior $FO(s)$ 4724	500 iterações Média $FO(s)$ 4675	307
Teste 2	Lista Tabu	Vizinhança	Condição de parada	para alcançar ótimo global
	10 Menor $FO(s)$ 4538	50 Maior $FO(s)$ 4786	500 iterações Média $FO(s)$ 4639	338
Teste 3	Lista Tabu	Vizinhança	Condição de parada	para alcançar ótimo global
	5 Menor $FO(s)$ 4664	120 Maior $FO(s)$ 4790	500 iterações Média $FO(s)$ 4742	500

A figura 9 ilustra a comparação dos resultados obtidos pelas técnicas AG e BT em relação à execução que obteve a melhor  $FO(s)$  no cenário 2, no caso do AG teste 2 execução 3 e no caso do BT teste 3 execução 5. A figura 10 ilustra a comparação, entre os algoritmos, no melhor teste de cada, neste caso segundo as médias das execuções por iteração.

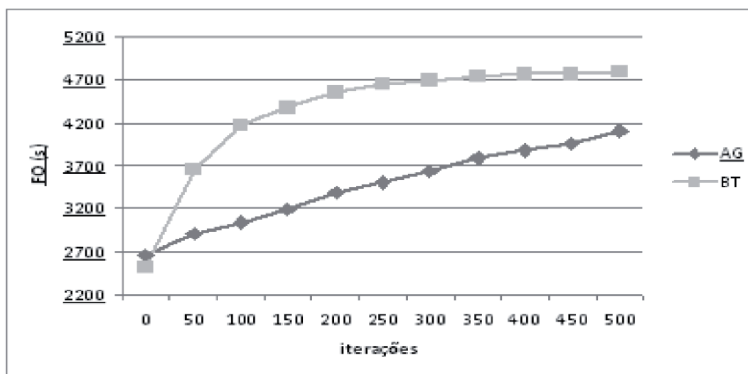


Figura 9: Comparação do desempenho das técnicas AG e BT no Cenário 2, segundo a melhor execução de cada

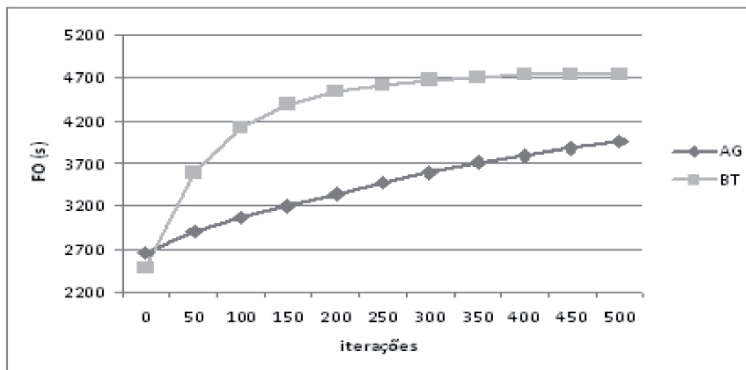


Figura 10: Comparação do desempenho das técnicas AG e BT no Cenário 2  
 Nota: teste 2 AG e teste 3 BT, segundo a média das execuções do melhor teste de cada

Ao analisar a figura 9, observa-se que as soluções encontradas pelo AG melhoraram consideravelmente, se comparadas as soluções encontradas no cenário 1. Este fato se explica pelas mudanças nos valores de mínimo e máximo de profissionais permitidos em um ST. Porém, apesar da melhora no desempenho do algoritmo, as restrições ainda existentes, referentes a preferências dos profissionais por ST, e o operador de *crossover* utilizado limitam a exploração do espaço de busca do AG.

Comparando os resultados do segundo cenário com o primeiro, não houve diferença considerável no desempenho da BT, a mesma sempre apresentou boas soluções em ambos os cenários.

Com base nos testes realizados em ambos os cenários, pode-se observar a superioridade do algoritmo BT para solucionar o problema tratado. A diferença entre o desempenho dos 2 algoritmos deve-se ao fato de que a BT parte de uma solução inicial factível e segue fazendo pequenas mudanças (vizinhança) para gerar outras soluções, que sempre serão factíveis. Ao contrário, o AG parte de um conjunto de soluções iniciais factíveis, porém, por meio do operador de *crossover* utilizado realiza a troca de blocos de genes entre duas soluções para gerar uma nova solução nem sempre factível.

## CONCLUSÃO

Através dos resultados obtidos e expostos na seção 6, observa-se que a utilização do método BT se mostra como uma alternativa

viável, boa e superior ao AG utilizado, para solucionar o problema tratado. O baixo desempenho obtido pelo AG explica-se pelas restrições existentes no problema abordado combinadas com o operador de cruzamento utilizado, que na maioria das vezes criava soluções ineficazes.

Como trabalhos futuros, sugere-se o uso de heurísticas construtivas para montar a solução inicial para os algoritmos, já que testes iniciais demonstraram um bom desempenho. Outra sugestão, é a implementação de uma ferramenta computacional baseada em um método híbrido, utilizando AGs e BT, para solucionar o problema do HU-Dourados.

## GENETIC ALGORITHMS AND TABU SEARCH IN THE THE BODY OF A UNIVERSITY HOSPITAL NURSING

*Abstract: the Nurse Scheduling Problem is an expensive and complex task, causing loss of precious time and attention of an important employee of the hospital, usually chief nurse. Thus, this paper proposes the application of the Genetic Algorithms (GA) and Tabu Search (TS) to resolve the Nurse Scheduling Problem of an University Hospital. The results obtained demonstrated the superiority of TS for solving the problem addressed.*

**Keywords:** Genetic Algorithms. Tabu Search. Timetabling. Nurse Rostering. Nurse. Metaheuristics.

### Referências

BARRA, T. V. Um Ambiente Evolutivo para Apoio ao Projeto de Antenas de Microfita. Dissertação (Mestrado) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2007.

BRUN, A. L. Algoritmos genéticos. In: Encontro Paranaense de Computação. [S.l.: s.n.], 2004.

CAMILO, C. G.; STELLE, D. Aplicando algoritmos genéticos ao problema de definição de escala de trabalho do corpo de enfermagem de um hospital universitário. In: XL SBPO. *Anais...*, 2008.

CIDADANIA, T. da. Grande Dourados - MS. 2008. [Online; acesso em 02-Agosto-2008]. Disponível em: <<http://www.territoriosdacidadania.gov.br>>.

DOWSLAND, K. Nurse scheduling with tabu search and strategic oscillation. *European Journal of Operational Research*, v. 106, p. 393–407, 1998.

FREITAS, C. C. et al. Uma ferramenta baseada em algoritmos genéticos para a geração de tabela de horário escolar. In: SÉTIMA ESCOLA REGIONAL DE COMPUTAÇÃO Bahia-Sergipe. Vitória da Conquista: [s.n.], 2007.

GLOVER, F. Tabu search: A tutorial. *Interfaces*, v. 4, n. 20, p. 74-94, 1990.

GLOVER, F.; LAGUNA, M. Tabu search. In: REEVES, C. (Ed.). *Modern Heuristic Techniques for Combinatorial Problems*. Oxford, England: Blackwell Scientific Publishing, 1993.

GROSKO, A. P.; GORSKI, J. R.; DIAS, J. da S. Algoritmo genético: revisão histórica e exemplificação. In: Sociedade Brasileira de Informática na Saúde. X Congresso Brasileiro de Informática na Saúde. Florianópolis - SC, 2006.

IKEGAMI, A.; NIWA, A. A subproblem-centric model and approach to the nurse scheduling problem. *Mathematical Programming*, v. 97, n. 3, p. 517–541, 2003.

LINDEN, R. Algoritmos Genéticos - Uma importante ferramenta da Inteligência Computacional. Brasport, 2006.

MARIANO, A. S. Geração automática de Grade Horária para a Faculdade de Engenharia Elétrica da UFU usando Algoritmos Genéticos. Dissertação (Mestrado Faculdade de Engenharia Elétrica) – Universidade Federal de Uberlândia, Uberlândia, 2007.

MARINHO, E. H. et al. Busca tabu aplicada ao problema de tripulações de Ônibus urbano. In: XXXVI - SBPO. São João del Rei: [s.n.], 2004.

MILLAR, H.; KIRAGU, M. Cyclic and non-cycling scheduling of 12h shift nurses by network programming. *European Journal of Operational Research*, v. 104, p. 582-594, 1998.

MOGNON, V. R. Algoritmos Genéticos Aplicados na Otimização de Antenas. Dissertação (Mestrado Faculdade de Engenharia Elétrica) – Universidade Federal do Paraná, Curitiba, 2004.

NETO, C. A. S.; CONSTANTINO, A. A.; ARAUJO, S. A. de. Ferramenta de apoio ao escalonamento de teleatendentes. *Espaço Energia*, v. 1, n. 5, Outubro 2006.

PAPPA, G. L. Seleção de atributos utilizando Algoritmos Genéticos multiobjetivos. Dissertação (Mestrado) – PUC-PR, Curitiba, 2002.

RANGEL, A. L.; ÉVORA, Y. D. M. Elaboração automática da escala periódica de trabalho dos profissionais de enfermagem por meio de um software específico. *Revista Eletrônica de Enfermagem*, v. 9, n. 1, p. 17–30, jan./abr. 2007.

Santos 2006. SANTOS, I. C. dos. O problema do escalonamento de enfermeiros. Departamento de Matemática, Março 2006.

SHIZEN; ZHOUYANG, C. T. Comparison of steady state and elitist selection genetic algorithms. In: IEEE. Proceedings of the 2004 International Conference

on Intelligent Mechatronics and Automation. Chengdu, China, 2004.

SOUZA, M. J. F.; MARTINS, A. X.; ARAÚJO, C. R. de. Experiências com simulated annealing e busca tabu na resolução do problema de alocação de salas. In: XXXIV - SBPO. Rio de Janeiro - RJ: [s.n.], 2002.

\* Recebido em: 02.03.2010

Aprovado em: 23.03.2010

#### CELSO CAMILO

Universidade Federal da Grande Dourados, Faculdade de Ciências Exatas e Tecnologia Dourados, MS, Brasil. *E-mail*: .celsocamilo@ufgd.edu.br.

#### DIOGO STELLE

Universidade Federal da Grande Dourados, Faculdade de Ciências Exatas e Tecnologia Dourados, MS, Brasil. *E-mail*: diogo.stelle@gmail.com.

#### DIEGO ROZA

Universidade Federal da Grande Dourados, Faculdade de Ciências Exatas e Tecnologia Dourados, MS, Brasil. *E-mail*: diego2607@gmail.com.