

EXTENSÃO DE UMA BUSCA LOCAL EXATA PARA O PROBLEMA DE ESCALONAMENTO DE ENFERMEIROS

Cristofer Alexandre Oswald (PIBIC/CNPq/FA/Uem), Ademir Aparecido Constantino (Orientador), e-mail: cristoferoswald@gmail.com

Universidade Estadual de Maringá / Centro de Tecnologia / Maringá, PR

Ciências Exatas e da Terra / Ciência da Computação

Palavras-chave: algoritmos heurísticos, busca local, problema de escalonamento

Resumo:

O Problema de Escalonamento de Enfermeiros é um problema de otimização que consiste em elaborar escalas de trabalho para os enfermeiros de um hospital, levando em consideração algumas restrições declaradas antecipadamente e buscando atender o máximo possível às preferências dos enfermeiros pelos turnos de trabalho. Este problema é classificado como NP-Difícil, assim, não existem algoritmos exatos que encontrem uma solução ótima para este problema. Para encontrar uma solução, uma maneira viável é a utilização de métodos heurísticos, que buscam encontrar soluções boas em tempo computacional viável, porém eles não garantem uma solução ótima. Este trabalho busca investigar um algoritmo heurístico para o Problema de Escalonamento de Enfermeiros baseado em Constantino et al. (2014). Para isso, utilizou-se de um método de busca local chamado *backtracking*, que almeja utilizar das várias soluções ótimas encontradas em um dia. O modelo implementado produziu bons resultados, com até 14% de melhora em média, mas por não garantirem todas as restrições, os resultados não podem ser considerados finais.

Introdução

Definir a escala dos enfermeiros para um hospital é muitas vezes uma tarefa custosa e que ocorre mensalmente. A atribuição de escalas não envolve apenas designar os turnos e dias de trabalho para um enfermeiro, necessita-se respeitar regras de demanda, leis trabalhistas, preferências dos empregados entre outras restrições. Realizar esse trabalho manualmente demanda tempo e, dependendo do tamanho do estabelecimento se torna muitas vezes inviável.

Problemas de escalonamento como esse estão presentes em muitas áreas práticas, como por exemplo, escalonamento de motoristas, veículos e cargas para determinadas rotas, distribuição de horários de aulas e matérias para professores, entre muitos outros.

Algoritmos eficientes em tempo que gerem soluções ótimas geralmente são inviáveis, assim busca-se implementar algoritmos utilizando heurísticas que gerem soluções aceitáveis e que tenham complexidade de tempo polinomial.

O problema de escalonamento de enfermeiros tem por objetivo construir um conjunto de escalas de trabalho para enfermeiros, atendendo a um conjunto de restrições. Elas podem ser divididas em dois grupos: flexíveis e rígidas. As rígidas precisam ser atendidas na sua integridade e são, por exemplo, o tempo mínimo de descanso entre os turnos ou máximo de horas trabalhadas por dia. Já as restrições flexíveis são desejáveis mas não estritamente necessárias e podem ser preferências de turnos e dias, por exemplo.

O algoritmo proposto por Constantino et al. (2014) para a resolução do problema de escalonamento de enfermeiros baseia-se numa fase construtiva, onde cada dia é resolvido individualmente. O objetivo desse trabalho é fazer um melhor uso das soluções de cada dia encontradas, já que, no trabalho original, é usada apenas a primeira solução encontrada. Para isso, foi implementado um algoritmo que utiliza busca local para procurar uma melhor solução na vizinhança de soluções iniciais.

Materiais e métodos

O método proposto é composto de três etapas principais: resolver o problema de designação de um dia, buscar as soluções ótimas e encontrar a melhor combinação de soluções ótimas.

Problema de designação de um dia

Para resolver o problema de designação para um dia, foi utilizado o algoritmo de Carpaneto e Toth (1987), que tem complexidade O^3 , chamado de *apc*. Esse algoritmo realiza sucessivas reduções na matriz de custos do dia, até que uma solução é encontrada.

A matriz de custos de um dia é dado pelas preferências do enfermeiro para cada turno daquele dia e, se uma restrição é quebrada ao designar um dado turno, é atribuído um custo proibitivo, assim, as restrições rígidas e as restrições flexíveis máximas são garantidas que não serão quebradas.

Uma solução é encontrada quando existe pelo menos uma combinação de colunas que tenham o valor 0 e que cubram todas as linhas. Uma das saídas do algoritmo é a matriz de custos reduzida, onde cada elemento zerado faz parte de uma solução ótima e é essa matriz que é utilizada como entrada para a próxima etapa.

Busca pelas soluções ótimas

A segunda etapa consiste em procurar pela matriz de custos reduzida as diversas soluções existentes, isso é, as combinações de elementos zerados. Visto que, no pior caso, onde todos os elementos são zeros, a complexidade é n^n foi colocado um parâmetro de parada k , ou seja, um máximo de soluções a ser procurado. Para os resultados aqui mostrados, esse valor foi de 4, assim, cada dia, 4 soluções vizinhas são computadas. É importante notar que as soluções de cada dia são ótimas e, para um mesmo dia, tem o mesmo custo, mas têm designações diferentes.

Busca pela melhor combinação de soluções

Essa etapa baseia-se no algoritmo de *backtracking*. Esse algoritmo cria uma árvore de soluções finais com base nas soluções encontradas de cada dia. Já que as soluções para um mesmo dia tem o mesmo custo mas tem designações diferentes, escolher uma solução ou outra pode levar a resultados diferentes nos dias seguintes. Assim, o algoritmo combina as diferentes soluções, dia a dia, procurando a solução final de menor custo. Novamente, esse algoritmo tem complexidade exponencial, de k^{nd} , onde nd é o número de dias, assim, foi definido uma profundidade máxima que o algoritmo poderia buscar na árvore. Para os resultados aqui apresentados, esse valor foi de 7, ou seja, a profundidade da busca foi de 7 dias. Ao chegar na profundidade máxima, o algoritmo dá uma solução para esses dias e reinicia a partir do próximo dia.

Abaixo, o algoritmo geral desenvolvido, com suas três etapas.

Tabela 1: algoritmo geral do modelo desenvolvido

ComputarDia(dia):

```
matriz_custos = dia.calcCustos()
matriz_reduzida = apc(matriz_custos)
soluções = buscarSoluções(matriz_reduzida)
```

Para cada solução em soluções:

```
dia.designarEnfermeiros(solução)
custo = ComputarDia(dia.proxímo)
```

Se custo < menor_custo:

```
removerSoluçãoFinal(menor_solução)
menor_custo = custo
menor_solução = solução
adicionarSoluçãoFinal(solução)
```

```
dia.desatribuirEnfermeiros()
```

retorne menor_custo

Para avaliação do método proposto, fui utilizado a biblioteca NSPLib, desenvolvido por Maenhout e Vanhoucke (2005), que consiste de diversas instâncias para o problema de escalonamento de enfermeiros. A biblioteca é dividida em duas categorias: “diverso” e “realista”. O primeiro conta com 7290 escalas de 7 dias para 25, 50, 75 e 100 enfermeiros. Já a segunda é composta por 960 escalas de 28 dias para 30 e 60 enfermeiros. Além disso, o NSPLib conta com 16 arquivos de caso, que contêm restrições diferentes para as possíveis escalas, sendo que as 8 primeiras são para o grupo “diverso” e as outras 8 para o grupo “realista”. Dessa maneira, tem-se um total de 248640 instâncias. O modelo foi avaliado utilizando todas as instâncias.

Resultados e Discussão

Inicialmente, desejou-se fazer uma comparação direta com o modelo proposto por Constantino *et al.* (2014), mas, para isso, seria necessário garantir que todas as restrições não fossem violadas, o que não foi possível implementar a partir dos algoritmos dados. Nesse trabalho, apenas as restrições rígidas e as flexíveis

máximas são garantidas, faltando as flexíveis mínimas. Constantino *et al.* (2014) realiza uma alteração final nas suas escalas de modo que todas as restrições são garantidas, ao não priorizar a demanda. Aqui, a demanda é totalmente coberta. Mesmo assim, foi realizada a comparação, visto que assim é possível avaliar se o método proposto é plausível e se trabalhos futuros que implementem essas alterações são possíveis.

Para a comparação dos resultados, foi feito uma média entre todas as instâncias de cada quantidade de enfermeiros (25, 50, 75, 100, 30 e 60). Os dados mostraram uma redução de custo significativa, em média, 14%, sendo a maior diferença de 21% para instâncias de 30 enfermeiros. A diferença é ainda maior quando considerado o tempo de computação, que foi 94% menor em média.

Conclusões

Com o objetivo de melhor utilizar as múltiplas soluções providas pelo *apc*, foi implementado um algoritmo que combinava as soluções, procurando a melhor combinação. Os resultados são promissores, mas, visto que as restrições não estão todas garantidas, não é possível tratar esses resultados como finais. Trabalhos futuros podem utilizar essa implementação como base e adicionar mecanismos de garantir todas as restrições, o que, com um bom método, podem prover resultados factíveis e melhores que o trabalho de Constantino *et al.* (2014). O baixo tempo de computação dos resultados um grande garante espaço de tempo para um futuro método de melhoramento das restrições. Por último, Constantino *et al.* (2014) implementa uma etapa de melhoramento das soluções iniciais, que, em conjunto com o modelo aqui implementado, pode trazer resultados ainda melhores.

Agradecimentos

À Fundação Araucária pelo apoio financeiro.

Referências

- Burke, E. K., Li, J., & Qu, R. (2012). **A Pareto-based search methodology for multi-objective nurse scheduling**. *Annals of Operations Research*, 196(1), 91–109.
- Constantino, Ademir Aparecido and Landa-Silva, Dario and de Melo, Everton Luiz and de Mendonza, Candido Ferreira Xavier and Rizzato, Douglas Baroni and Romao, Wesley (2014). **A heuristic algorithm based on multi-assignment procedures for nurse scheduling**. *Annals of Operations Research*, 218 (1). pp. 165-183. ISSN 1572-9338
- Carpaneto, G., & Toth, P. (1987). **Primal-dual algorithms for the assignment problem**. *Discrete Applied Mathematics*, 18, 137–153.
- Maenhout, B., & Vanhoucke, M. (2005). **NSPLib—a Nurse Scheduling Problem Library: a tool to evaluate(meta-)heuristic procedures**. In *O.R. in health* (pp. 151–165). 31st meeting of the EURO working group on OR applied to health services. Amsterdam: Elsevier.