

Projeto de Escalonamento

Cecilia Fernandes
Lucas Cavalcanti dos Santos

Prof. Alfredo Goldman

17 de dezembro de 2009

Sumário

1	Introdução	2
2	Motivação	3
3	Heurísticas	4
3.1	First Fit	4
3.2	First Fit Decreasing	5
3.3	Next Fit	5
3.4	Best Fit	6
3.5	Best Fit Decreasing	6
3.6	Worst Fit	7
4	O programa	8
5	Mais restrições e heurísticas	9
6	O futuro	10
7	Conclusão	11

1 Introdução

Um escalonamento é uma forma de dividir a execução de tarefas em diferentes recursos.

É um conceito que, no contexto de linhas de produção fabris, é usado para diminuir custos de matéria-prima ou para aumentar a velocidade ou capacidade de produção. Dadas tarefas que máquinas sabem executar, como dividi-las de forma que essas restrições sejam atendidas de maneira eficiente.

Em computação, o escalonamento de tarefas é usualmente associado à divisão do tempo de um processador ou mesmo na divisão de processos em diferentes *cores* de máquinas com mais de um processador. Ainda há a extensa e vital utilização de escalonamento na paralelização de tarefas em diferentes máquinas.

Cada problema de escalonamento tem características específicas e, portanto, heurísticas e algoritmos diferentes que se aplicam melhor a cada um.

Outros trabalhos também desenvolvidos nessa matéria estudaram as diversas heurísticas, meta-heurísticas e restrições aplicáveis a problemas de escalonamento. Nesse trabalho, criamos a infraestrutura para que seja fácil implementar tais heurísticas e vê-la funcionando na prática.

De modo lúdico, com uma motivação para o problema, abordaremos as formas de escalonar programas de televisão de um canal de forma a atender às necessidades ou desejos de uma família tida como “modelo”, apresentada ao leitor na seção de Motivação.

Em seguida, mostraremos as Heurísticas implementadas explicando a idéia do algoritmo e mostrando *screenshots* dos escalonamentos gerados por essas heurísticas para uma determinada entrada do problema.

Depois, vêm as seções de Desenvolvimento, onde explicamos como a ferramenta desenvolvida é facilmente extensível para implementação de novos escalonamentos, e de Outras Restrições que poderiam ser adicionadas ao problema de modo a deixá-lo mais realista e interessante.

Finalmente, os Passos Futuros em direção aos quais o projeto pode caminhar são mencionados e a conclusão que tiramos com relação às heurísticas implementadas e ao problema abordado.

2 Motivação

Motivação para codar, personagens e problema mais simples

3 Heurísticas

O problema de escalonar programas em uma grade de programação de uma emissora de TV a cabo é um caso particular de um problema bastante conhecido de teoria da combinatória, e que também pode ser visto como um problema de escalonamento: o *Bin-packing*.

Esse problema é conhecidamente NP-difícil, então não é possível conseguir a solução ótima em um tempo razoável. Por causa disso, ao invés de tentarmos descobrir a solução ótima, temos que ir atrás de uma solução que é apenas satisfatória, uma aproximação. Para conseguir uma aproximação precisamos usar heurísticas.

Heurísticas são algoritmos de aproximação, ou seja, algoritmos que não buscam encontrar a solução ótima, mas uma solução que consiga se aproximar da ótima segundo algum fator. Esse fator é a razão entre a solução aproximada e a ótima, significando o quão boa é a aproximação.

Uma característica importante de heurísticas é que elas tomam decisões específicas quando chegam em determinado ponto do algoritmo. Por exemplo escolher o maior item, inverter a ordem de itens da solução, etc.

Para conseguir uma solução aproximada para o problema *Bin-packing* podemos utilizar várias heurísticas. Algumas delas foram implementadas e rodadas com a mesma entrada.

3.1 First Fit

O *First Fit* coloca o item no primeiro bin em que ele cabe. No caso do problema da grade de programação, o programa é colocando no primeiro dia em que ele cabe. Caso o programa não caiba em nenhum dos dias, é criado um novo dia.

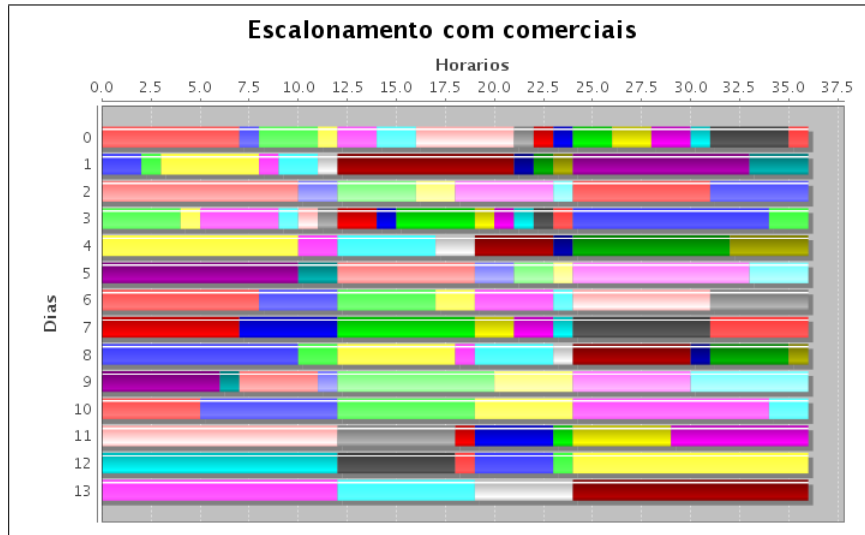


Figura 1: Resolução de um problema usando o First Fit

3.2 First Fit Decreasing

O *First Fit Decreasing* é uma variação do First Fit em que os itens são ordenados em ordem decrescente de valor antes de rodar o algoritmo. Essa heurística só pode ser usada se todos os itens forem conhecidos de antemão.

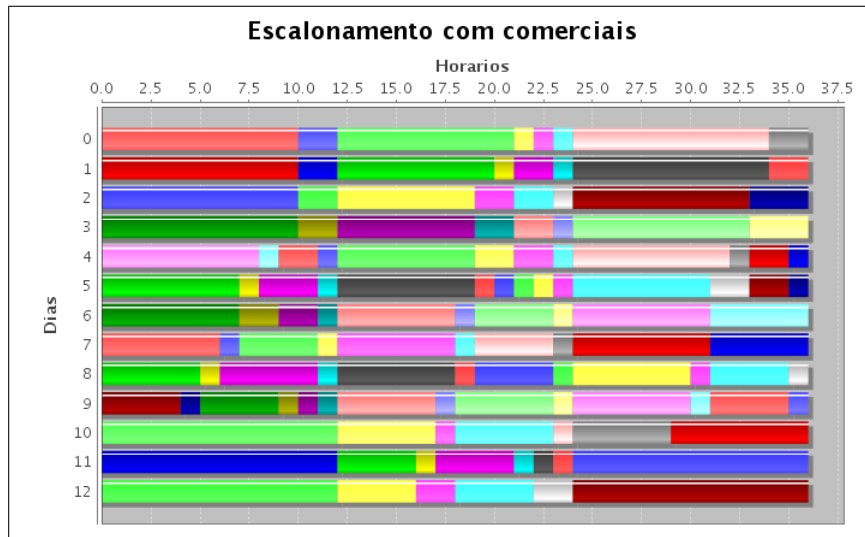


Figura 2: Resolução de um problema usando o First Fit Decreasing

3.3 Next Fit

O *Next Fit* tenta preencher um bin usando o primeiro item que caiba, se nenhum programa couber, um novo bin é criado. A principal diferença entre o *Next Fit* e o *First Fit* é que no primeiro é preenchido um bin por vez e no segundo tenta-se colocar o item no primeiro bin que ele caiba.

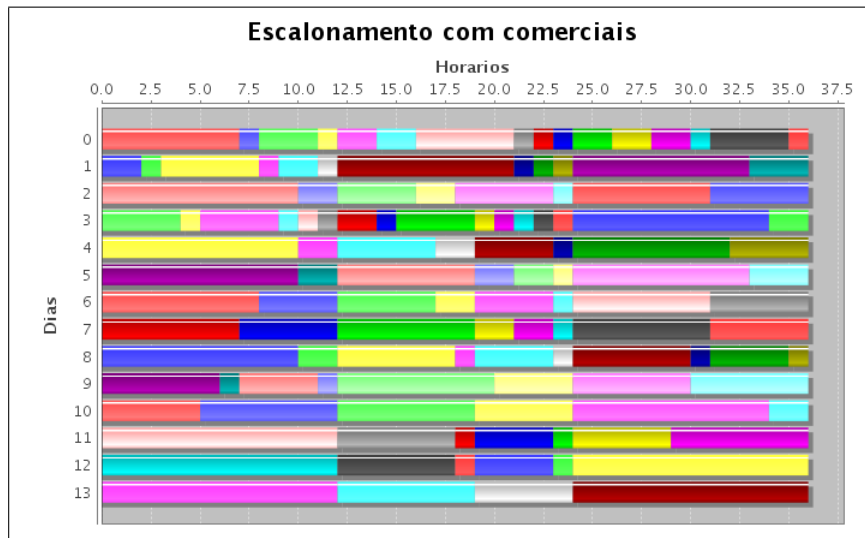


Figura 3: Resolução de um problema usando o Next Fit

3.4 Best Fit

O *Best Fit* tenta colocar o item no bin em que, após colocar o item, sobre o mínimo de espaço restante no bin. Caso não caiba em nenhum bin, é criado um novo.

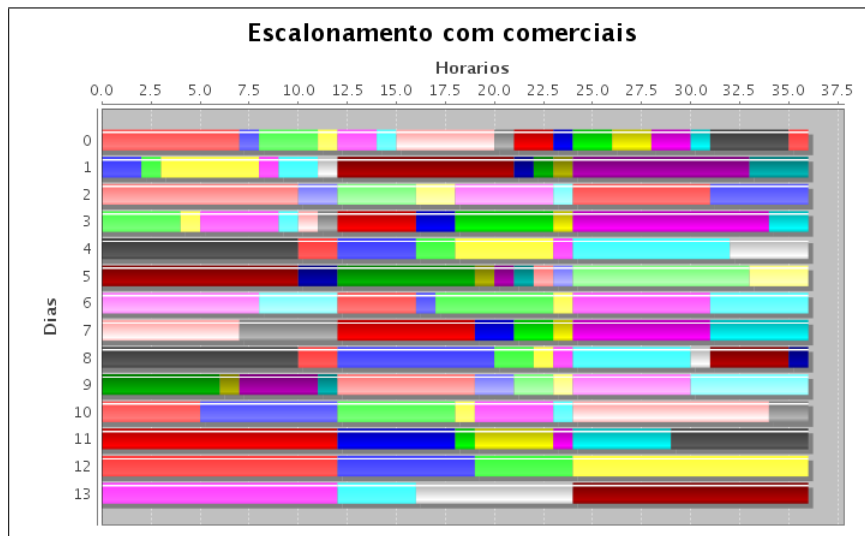


Figura 4: Resolução de um problema usando o Best Fit

3.5 Best Fit Decreasing

O *Best Fit Decreasing* é uma variação do *Best Fit* em que os itens são ordenados em ordem decrescente de valor antes de rodar o Algoritmo.

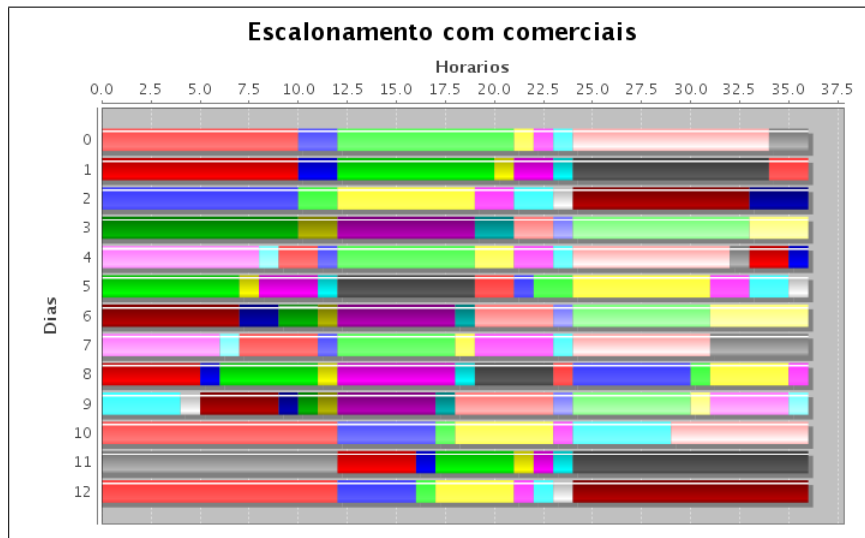


Figura 5: Resolução de um problema usando o Best Fit Decreasing

3.6 Worst Fit

O *Worst Fit* tenta colocar o item no bin em que, após colocar o item, sobre o máximo de espaço restante no bin. Caso não caiba em nenhum bin, é criado um novo.

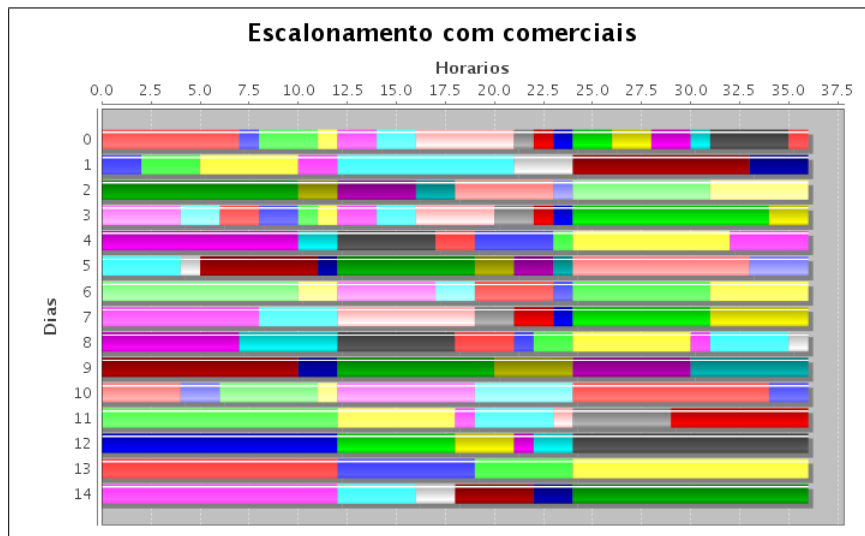


Figura 6: Resolução de um problema usando o Worst Fit

4 O programa

5 Mais restrições e heurísticas

Reservas e outras restrições ou heurísticas que podiam ser implementadas

6 O futuro

quais as próximas coisas que poderiam ser implementadas.

7 Conclusão

Referências

- [1] Autor (ano). *Nome da publicação*. Editora, edição;