

MAC0461  
Introdução ao Escalonamento e Aplicações  
Prof. Alfredo Goldman vel Lejbman  
Escalonamento em linha de produção

J.P. Kerr Catunda #USP:2961692



Instituto de Matemática e Estatística - USP  
Departamento de Ciência da Computação

17 de novembro de 2009

## Sumário

<b>1</b>	<b>Objetivo</b>	<b>3</b>
<b>2</b>	<b>O problema</b>	<b>3</b>
2.1	A linha de produção . . . . .	3
2.1.1	A esteira . . . . .	3
2.1.2	Particularidades da linha . . . . .	3
2.2	O estoque . . . . .	4
2.2.1	Particularidades do estoque . . . . .	4
2.3	A demanda . . . . .	4
2.3.1	Particularidades da demanda . . . . .	4
2.4	O Desafio . . . . .	4
<b>3</b>	<b>Modelagem</b>	<b>4</b>
3.1	Motivação . . . . .	5
3.2	Heurística . . . . .	5
3.2.1	Parte geométrica . . . . .	5
3.2.2	Parte temporal . . . . .	5
<b>4</b>	<b>Parte geométrica</b>	<b>5</b>
4.1	Figuras . . . . .	5

4.1.1	Moldes . . . . .	5
4.1.2	Demanda . . . . .	6
4.1.3	Turnos . . . . .	6
4.2	Capacidade da linha de produção . . . . .	7
4.3	Resolvendo . . . . .	8
4.3.1	<i>First come first serve</i> . . . . .	8
4.3.2	<i>Bin Packing</i> . . . . .	8
4.3.3	Uma nova perspectiva: Olhando os turnos em paralelo . . . . .	9
4.3.4	<i>Round Robin</i> - Paralelizando sem critério . . . . .	9
4.3.5	<i>Meta Round Robin</i> . . . . .	10
4.3.6	<i>Meta Round Robin</i> parcial . . . . .	11
4.3.7	Combinando soluções . . . . .	11
<b>5</b>	<b>Parte temporal</b>	<b>12</b>
5.1	Um olhar cuidadoso sobre demanda . . . . .	12
5.2	Projeção de estoque . . . . .	13
5.3	Projeção de estoque com retirada de lotes . . . . .	13
5.4	A ordem dos turnos influencia? . . . . .	13
5.5	Motivação: Séries de Fourier . . . . .	13
5.5.1	Comportamento de diferentes projeções de estoque . . . . .	14
5.6	Parte Temporal influenciando a parte Geométrica . . . . .	14
5.7	Uma nova heurística . . . . .	15
5.8	A solução temporal . . . . .	16
<b>6</b>	<b>Conclusão</b>	<b>17</b>
6.1	Resultados . . . . .	17
6.2	Considerações sobre eficiência . . . . .	18

# 1 Objetivo

Queremos resolver um problema de escalonamento em uma fábrica. Ele deve satisfazer várias restrições tanto da linha de produção quanto de logística/estoque. Para isso modelaremos ele com ferramentas vistas durante o curso de MAC0461 e desenvolveremos um programa para aplicar e mostrar as soluções encontradas.

## 2 O problema

Uma fábrica fornece peças para diversos clientes. Para isso ela conta com:

- uma linha de produção
- um estoque
- pedidos de clientes
- um aluno do prof. Alfredo Goldman (Você!)

Existem muitas restrições a serem levadas em consideração. Não apresentaremos todas agora pois isso não nos ajudará a resolver o problema. Ao invés disso iremos começar com um problema mais simples e iremos refinando ele.

### 2.1 A linha de produção

#### 2.1.1 A esteira

A linha de produção possui uma esteira circular onde moldes são afixados. Esta esteira passa por diversos setores onde seu conteúdo é preparado. Por último ela atravessa um forno. As peças são então retiradas de seus moldes e o processo recomeça.

#### 2.1.2 Particularidades da linha

- Não cabem todos os moldes na linha de uma vez. Por isso devemos parar a linha de tempos em tempos para trocar alguns moldes. Esta troca custa muito tempo. Teremos que tomar algum cuidado aqui para minimizar este tempo.
- A esteira funciona em turnos. Estes turnos coincidem com os turnos de funcionários. Por exemplo: três turnos por dia de aproximadamente 5 horas cada.
- Uma vez ligada, esteira anda devagar e sempre, dando algumas voltas por turno. A cada volta ela produz uma peça com cada molde da linha.

Por exemplo: Um turno com 30 voltas produz 30 peças de cada molde na linha por turno.

## 2.2 O estoque

Nossa fábrica conta com um estoque. Ele é importante por vários motivos. Em particular, as peças precisam curar ao sair do forno por quase um dia antes de serem entregues. Além disso, como não cabem todos os moldes na linha, temos que estocar algumas peças que não estão sendo produzidas para atendermos a demanda.

### 2.2.1 Particularidades do estoque

- Ele é pequeno
- as peças ocupam uma unidade de estoque, independente de seu tamanho

## 2.3 A demanda

A demanda é feita com certa antecedência. Seria muito legal se ela fosse fixa pois poderíamos criar uma solução cíclica ótima para o problema e ir pra casa. Entretanto a demanda depende de quantos carros são vendidos, o que depende de muita coisa e está além do poder das montadoras.

### 2.3.1 Particularidades da demanda

- É feita com uma semana de antecedência (pelo menos)
- Muda muito
- É maior que a capacidade da fábrica. *Sempre.*

## 2.4 O Desafio

Com a infraestrutura acima, precisamos produzir o maior número de peças possível.

# 3 Modelagem

Queremos otimizar em nosso problema duas coisas bem distintas: a disposição de moldes na linha de produção e o aproveitamento do estoque. Os problemas são concorrentes. Entretanto, se tentarmos resolver tudo de uma vez não chegaremos a lugar nenhum.

### 3.1 Motivação

Na literatura de sistemas físicos, uma forma de resolver um problema de condições de contorno com dependência temporal é utilizamos o método de separação de variáveis. Ele se resume em reescrever uma função  $f(x, t)$  da seguinte forma:  $f(x, t) = g(x) \cdot h(t)$ . O resultado disso é uma  $g(x)$  que nos dá a parte espacial e uma  $h(t)$  que transforma  $g(x)$  com o tempo.

Nosso problema não se encaixa nos moldes acima. Ele possui uma natureza combinatória e discreta, enquanto o acima possui uma natureza diferencial e contínua. Mas... Podemos aproveitar a idéia.

### 3.2 Heurística

Queremos quebrar nosso problema em uma parte geométrica e uma parte temporal.

#### 3.2.1 Parte geométrica

Queremos encontrar configurações que aproveitam o espaço da linha da melhor forma possível. Isto consiste em vários escalonamentos de moldes, um para cada turno, de forma que juntos atendam a demanda semanal.

#### 3.2.2 Parte temporal

Deveremos então encontrar a melhor ordem de se executar tais turnos para otimizar o estoque/logística.

## 4 Parte geométrica

### 4.1 Figuras

Nosso programa otimiza e desenha nossa linha de produção. As figuras utilizadas aqui serão geradas por ele. Vamos então explicar os desenhos aqui utilizados.

#### 4.1.1 Moldes

Moldes são representados por retângulos. O comprimento do retângulo é proporcional ao comprimento do molde. Moldes iguais possuem mesma cor.

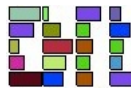


Figura 1: Alguns moldes

### 4.1.2 Demanda

A demanda é um conjunto de moldes que deve ser distribuído pelos turnos.

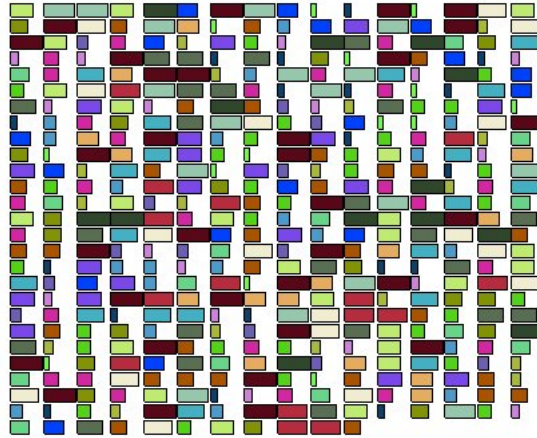


Figura 2: Um exemplo de demanda

### 4.1.3 Turnos

Representaremos um turno como um eixo cartesiano.



Figura 3: Um turno vazio

É legal ter em mente que a esteira é cíclica. Para isso temos uma marca no final de cada turno indicando o ponto onde ele retorna ao começo.

Um turno será preenchido com moldes. Moldes iguais possuem mesma cor.



Figura 4: Um turno com alguns moldes

Um turno só não faz verão. Uma semana possuirá alguns turnos.



Figura 5: 15 turnos relativos a uma semana de 5 dias úteis com 3 turnos por dia

## 4.2 Capacidade da linha de produção

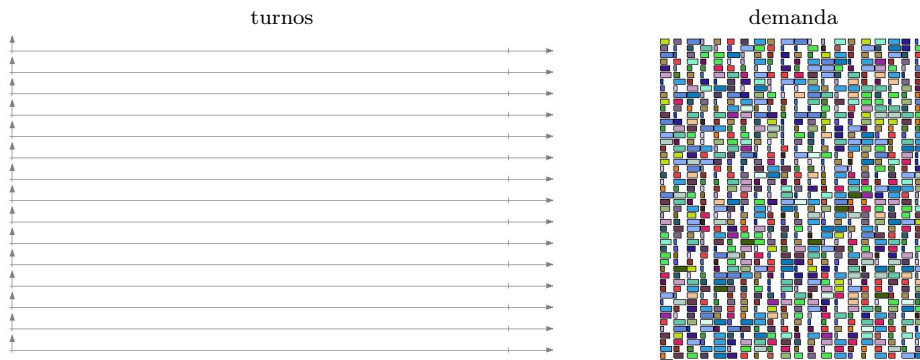
Sabemos que a demanda é sempre maior que a capacidade da fábrica. Devemos então estimar a capacidade da fábrica para não nos comprometer a fabricar mais do que conseguimos.

As informações importantes para isso são:

1. *Comprimento* da esteira
2. *Voltas* que a esteira dá por turno
3. *Turnos* que temos em uma semana

Ao olharmos o comprimento total percorrido pela esteira em uma semana teremos um *Upper Bound* (limitante superior)  $UB$  da capacidade da linha pois sabemos que a soma do comprimento de todos os moldes não pode ultrapassar este valor.

- $UB = \text{Comprimento} \times \text{Voltas} \times \text{Turnos}$
- Comprimento total de moldes:  $MoldeTot = \sum_{i=1}^n \text{Comprimento}(Molde_i)$
- $MoldeTot \leq UB$



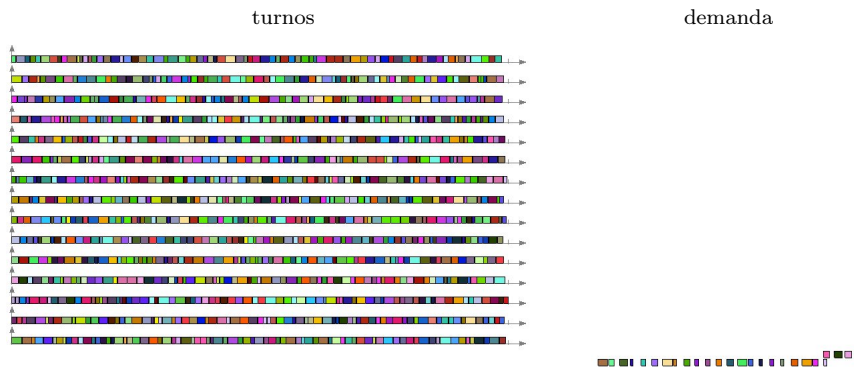
Linha de 93 metros com 15 turnos de 30 voltas - o exemplo que usaremos ao longo deste documento

### 4.3 Resolvendo

Vamos agora passar por diversas estratégias procurando resolver a parte geométrica a contento.

#### 4.3.1 *First come first serve*

Podemos preencher os turnos na ordem que as peças foram encomendadas.



Turnos gerados a partir da heurística *first come first serve*

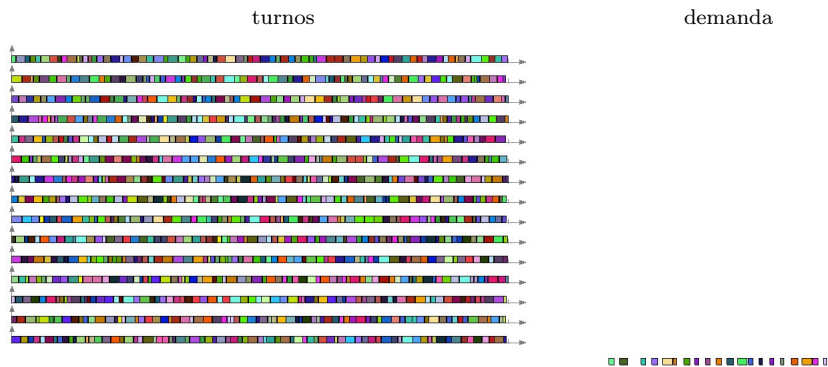
Este resultado não é nada interessante pois:

- Não tentamos preencher os turnos o máximo possível
- Não nos preocupamos em otimizar trocas

#### 4.3.2 *Bin Packing*

Podemos preencher os turnos com o algoritmo *bin packing* para ocupar o espaço da linha da melhor forma possível.





Turnos gerados a partir do *bin packing*

Este resultado, sozinho, também não é interessante pois não se preocupa em otimizar trocas na linha.

#### 4.3.3 Uma nova perspectiva: Olhando os turnos em paralelo

Até agora olhamos nosso problema como várias tarefas a serem executadas por uma máquina.

Na linguagem vista em sala, nosso problema seria:

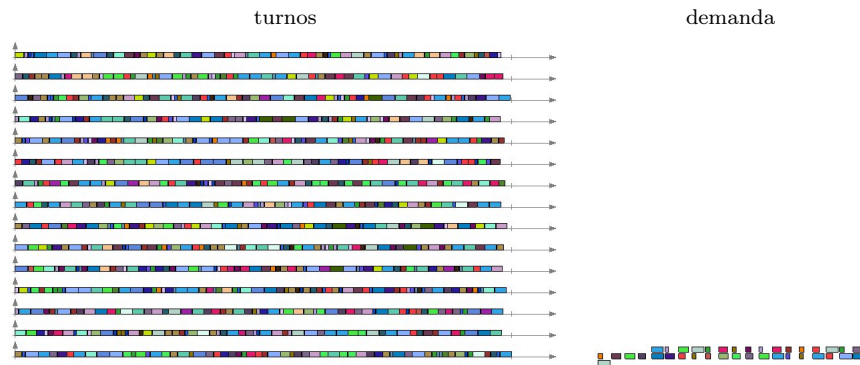
$$P | \max\{Tarefas\}$$

Se, ao invés disso olharmos cada turno como uma máquina, poderemos alocar as tarefas em diferentes turnos. Essa idéia simples transforma nosso problema em um problema paralelizável:

$$P_m | \max\{Tarefas\}$$

#### 4.3.4 Round Robin - Paralelizando sem critério

Com a idéia acima podemos usar algoritmos de atribuição de tarefas paralelas como, por exemplo, o *Round Robin*.



Turnos gerados a partir do *Round Robin*

Assim como as soluções anteriores, esta solução ainda não é interessante pois

- Não tentamos preencher os turnos o máximo possível
- Não nos preocupamos em otimizar trocas

Entretanto uma pequena variação no round robin pode nos ajudar com as trocas na linha.

#### 4.3.5 *Meta Round Robin*

Queremos construir soluções parecidas para minimizar as trocas na linha. Para isso, é razoável quereremos alocar moldes exatamente nas mesmas posições em turnos consecutivos.

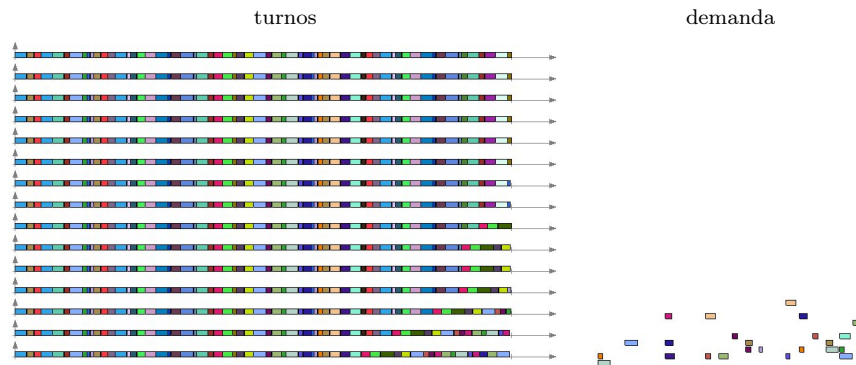
Para isso podemos alterar o Round Robin para que ele aloque moldes iguais em todos os turnos.

*Round Robin* modificado:

```

1 MetaRoundRobin() {
2   enquanto houver demanda
3     Pega molde com maior demanda
4     coloca um em cada turno
5     subtrai a demanda do molde em questão
6   Caso contrário
7     encerra
8 }
```

Apelidaremos carinhosamente esta variação de *meta Round Robin*. Ele consiste em, a cada iteração, organizar os moldes por demanda e alocar o mesmo molde uma vez em cada turno.



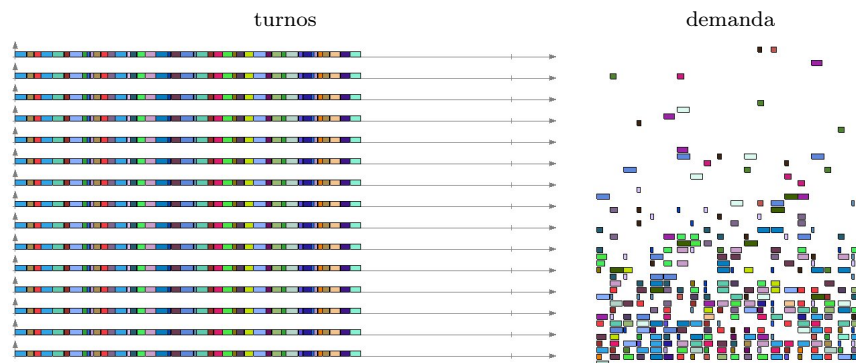
Turnos gerados a partir do *meta Round Robin*

Este resultado é interessante pois cria diversas configurações de linha com uma parte grande em comum, e isso é valioso por minimizar drasticamente o tempo de trocas. Entretanto este resultado deixa a desejar no preenchimento do resto da linha.

#### 4.3.6 *Meta Round Robin* parcial

Podemos aproveitar um pedaço de cada solução acima para construir algo melhor. Pensando desta forma temos, por hora, duas coisas boas para juntar:

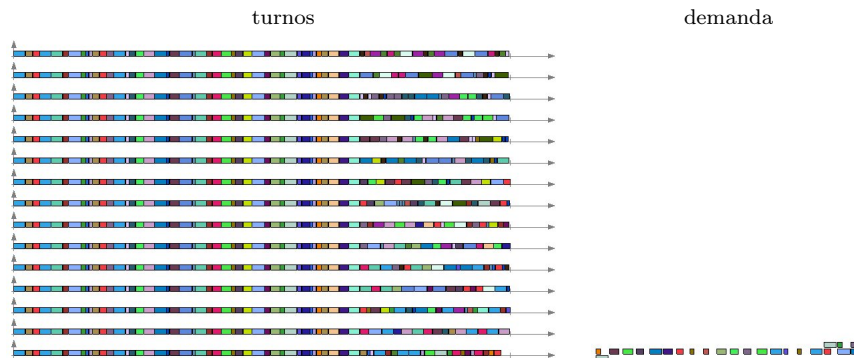
1. Preencher com *Meta Round Robin* enquanto conseguirmos repetir moldes em todos os turnos
2. Adotar alguma outra heurística para preencher o resto, possivelmente com o auxílio do *bin packing* para evitar desperdício de espaço.



Turnos gerados a partir do *meta Round Robin* parcial

#### 4.3.7 Combinando soluções

É interessante combinar soluções. Com isso poderemos aproveitar o melhor de cada uma delas.



Turnos gerados a partir do *Meta Round Robin* parcial seguido do bin packing

Mas qual seria a melhor combinação de soluções? Vamos parar com a solução geométrica por aqui e pensar um pouco sobre a solução temporal.

## 5 Parte temporal

Criamos até agora turnos com algumas propriedades geométricas interessantes.

Lembrando que nossa proposta é encontrar a solução geométrica e a solução temporal, precisamos ver como seria esta solução temporal.

A solução geométrica são turnos com as propriedades desejadas. Esta solução garante que a produção seja a maior possível e atenda a demanda. Para todos os efeitos, estes turnos podem ser executados em qualquer ordem.

Restam várias dúvidas:

- Como será que estes turnos se comportam se colocarmos ele para funcionar na prática?
- Como eles irão interagir com a demanda e o estoque?
- Será que existe alguma propriedade geométrica que influencia de alguma forma construtiva na parte temporal?

Agora nossa solução temporal se preocupará em colocar estes turnos em uma ordem tal que as restrições temporais, tais como peças em estoque, sejam respeitadas.

### 5.1 Um olhar cuidadoso sobre demanda

Até agora olhamos para a demanda como sendo semanal. Isto é verdade, mas não só. As montadoras não fazem questão de receber todas as peças de uma vez. Eles retiram as peças em lotes que correspondem as suas necessidades.

Com isso, ao produzirmos um lote, podemos despachá-lo para a montadora. Isso é importante para liberarmos espaço em nosso estoque.

## 5.2 Projeção de estoque

Podemos fazer uma projeção de estoque para vermos o que acontecerá com ele ao colocarmos a linha de produção para rodar.

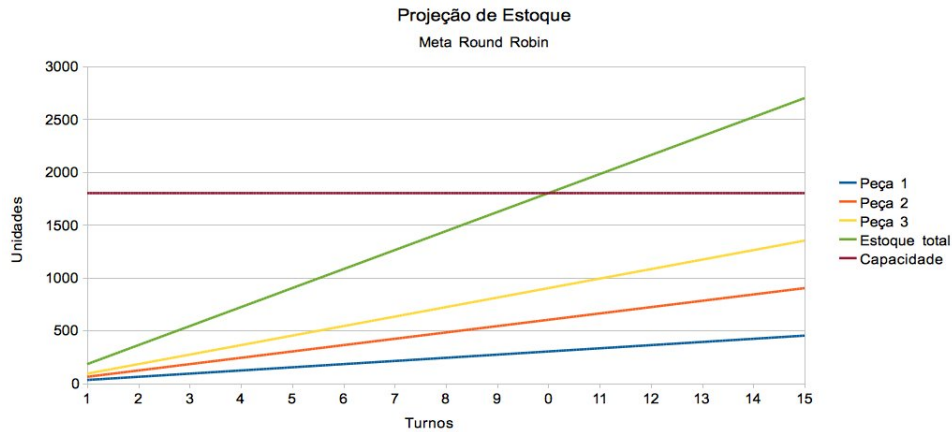


Figura 6: Gráfico de projeção de estoque em função de voltas da linha

Neste gráfico vemos a quantidade de peças em estoque, o total ocupado pelas peças e a capacidade total do estoque em função dos turnos

## 5.3 Projeção de estoque com retirada de lotes

A projeção de estoque acima não leva em consideração a retirada de lotes. Entretanto o estoque de cada peça deve dar uma caída cada vez que entregamos um lote de peças.

## 5.4 A ordem dos turnos influencia?

Ao quebrarmos o problema em parte geométrica e temporal tínhamos uma proposta bem clara: Alocar o máximo possível na linha com a parte geométrica e intercalar os turnos de uma forma inteligente na parte temporal. Mas... A ordem realmente influencia? *Quando* a ordem influencia?

## 5.5 Motivação: Séries de Fourier

Nossa projeção de estoque é cíclica. Ao sobrepormos várias delas, teremos o status atual de nosso estoque.

Seria fantástico se conseguíssemos somar elas e conseguir uma função que ficasse abaixo da capacidade máxima do estoque, assim como uma série de Fourier de uma função constante. Entretanto uma série de Fourier conta com senos e cossenos de diferentes frequências, e nós não temos como escolher a frequência de nossas projeções.

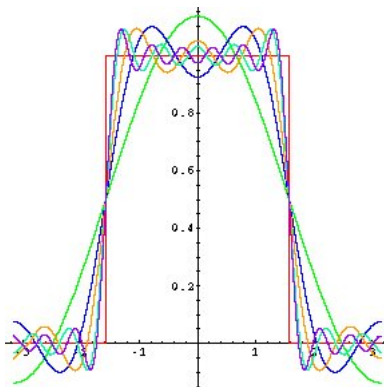
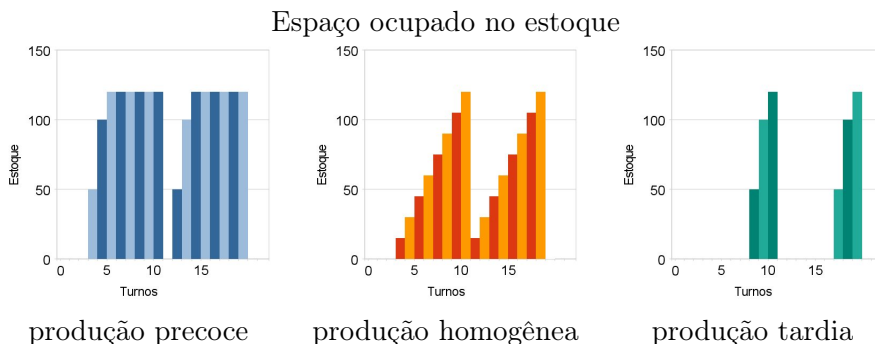


Figura 7: Convergência da série de Fourier de uma função degrau.

Mesmo que não seja possível construir as projeções de estoque como termos de uma série de Fourier, vamos fazer o possível para intercalá-los de forma a minimizar o total de estoque ocupado.

### 5.5.1 Comportamento de diferentes projeções de estoque

As peças, depois de produzidas, ficarão no estoque até serem entregues. Podemos ver nos gráficos abaixo o impacto no estoque da produção de 120 peças de diferentes formas:



Olhando acima vemos que, para nosso estoque, o melhor é deixar para produzir o maior número de peças em cima da hora.

Entretanto peças com uma demanda alta deverão ser produzidas em todos os turnos pois não temos moldes o suficiente para produzir todas de uma vez, sem contar que se tivéssemos, teríamos muitas trocas na linha.

## 5.6 Parte Temporal influenciando a parte Geométrica

Nosso *meta Round Robin* se encaixa na produção homogênea. Apesar de não parecer ótimo, não temos muito como fugir dele pois:

1. Temos uma demanda diária. Colocar vários moldes iguais na linha antes da entrega implica em muitas trocas de molde, o que toma muito tempo
2. Não temos muitos moldes iguais. Portanto, mesmo que as trocas não fossem um problema, não temos moldes suficientes para isso.

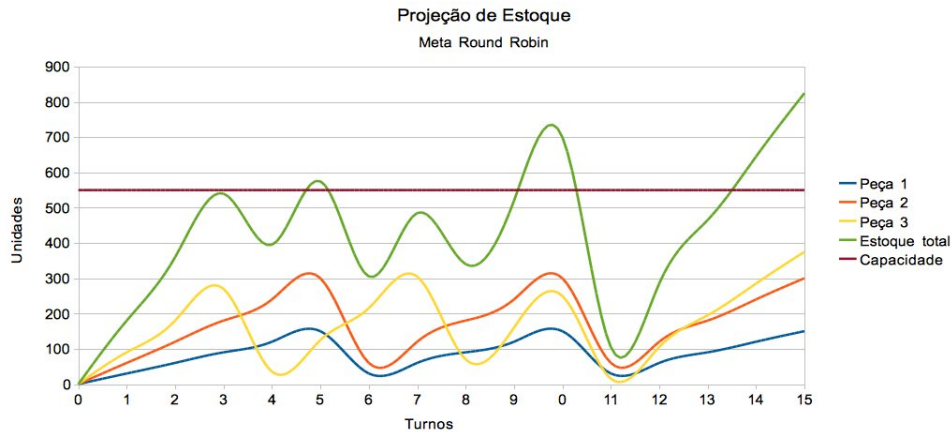


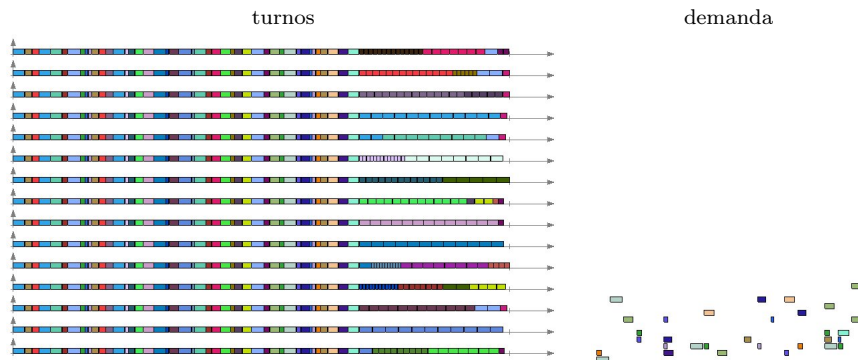
Figura 8: Gráfico de projeção de estoque em função de voltas da linha com retirada de lotes para o meta round robin.

Temos ainda outro problema: se turnos forem iguais, nossa modelagem temporal definitivamente não funcionará. Que vantagem teremos em trocar a ordem de dois turnos iguais?

Entretanto, com isso em mente e as projeções de estoque acima, vemos que a solução temporal está nos sugerindo uma nova heurística para preencher os turnos.

## 5.7 Uma nova heurística

Podemos aproveitar o *meta Round Robin* parcial e preencher a linha concentrando moldes iguais ao final de cada turno.



Turnos gerados a partir do *Meta Round Robin* parcial seguido da nova heurística

Isso nos dará um pico de produção das peças repetidas naquele turno.

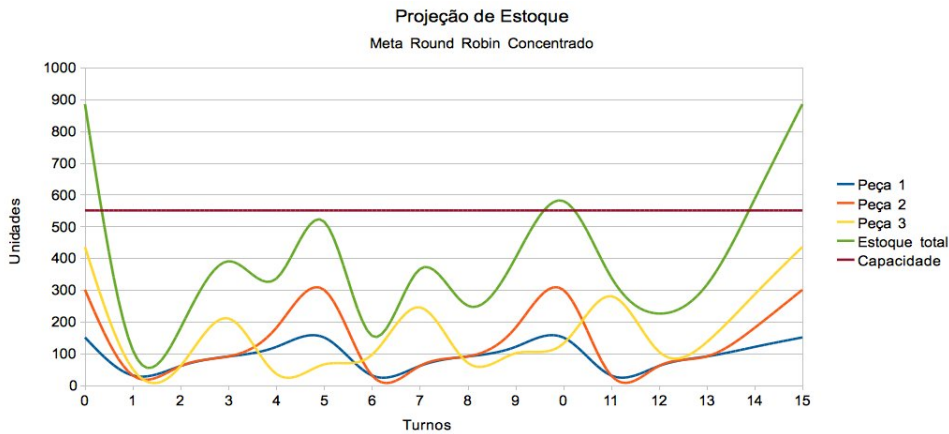


Figura 9: Picos de produção obtidos ao concentrar peças iguais depois do *meta Round Robin*

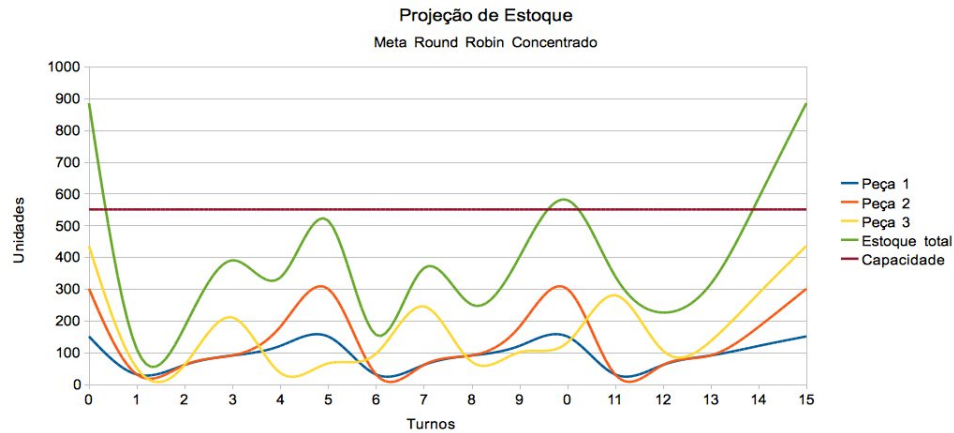
### 5.8 A solução temporal

Agora sim podemos tirar vantagem do escalonamento de turnos. Temos apenas a produção mínima necessária até em cima da hora, quando finalmente produzimos o máximo de peças repetidas por turno. Tudo que precisamos fazer para isso agora é tirar os picos de fase. Para isso devemos ter diferentes estoques iniciais para cada peça, fazendo com que as entregas de lotes estejam fora de “fase”.

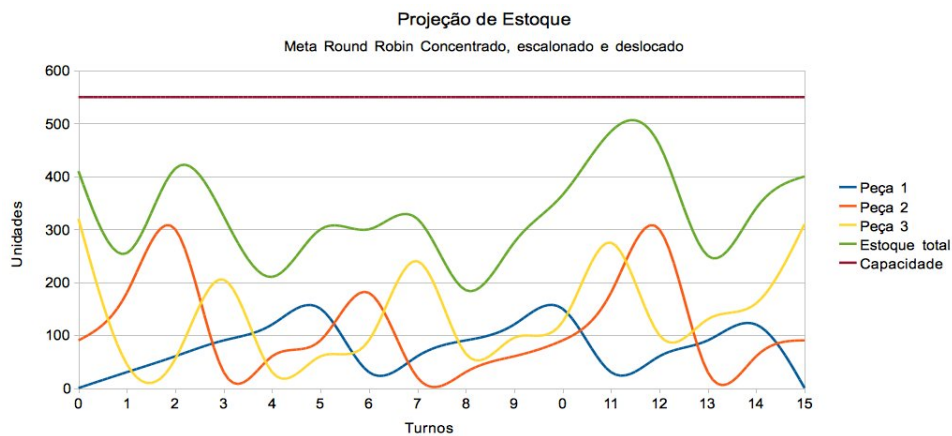


## Projeções de estoque antes e depois do escalonamento de turnos

### Antes de escalonar



### Depois de escalonar



Podemos conseguir isso forçando um excesso de produção das peças que queremos adiantar a fase. Desta forma, os lotes ficarão prontos de acordo com a nova fase de estoque daquela peça.

Note que aqui os gráficos não mais começam do zero, o que faz sentido. Uma semana é aproximadamente cíclica e, para conseguirmos o deslocamento de fase, temos que contar com um estoque inicial. Dada a natureza cíclica do problema, o estoque no fim da semana é também o do começo.

## 6 Conclusão

### 6.1 Resultados

Obtivemos sucesso nos dois pontos que nos propusemos: Minimizamos as trocas de moldes na linha e minimizamos o o estoque, mantendo ele dentro do limite estipulado.

## 6.2 Considerações sobre eficiência

A parte geométrica do problema pode ser resolvida da melhor forma possível (*bin packing*) por conta do número restrito de peças.

Caso o número de peças se torne grande a ponto de não podermos encontrar uma solução ótima (o que não é difícil), um algoritmo guloso pode ser colocado em seu lugar.

Note que a solução geométrica ótima não garante que a solução do problema é ótima. A parte temporal é complexa por si só e não existem garantias que o resultado encontrado é ótimo. Ao final, utilizaremos *bin packing* apenas para preencher a linha de produção depois de usarmos o *Meta Round Robin* parcial e concentrar peças iguais nos turnos. Como a demanda é maior que a capacidade da planta, os dois primeiros métodos já preenchem os turnos, e o *Bin packing* acaba por ser pouco utilizado.