



Relatório

ESTUDO DO FUNCIONAMENTO DE UMA FÁBRICA DE SKATES - SCC

Etiandro Cirilo André António – 2017290285

O modelo foi implementado na linguagem de programação Python com a biblioteca Salabim versão 19.0.2, para os processos uso o método Resource presente no salabim, para as filas de espera é usado o método Queue também presente no Salabim e para a coordenação das horas de trabalho, o método State.

Funcionamento do modelo

A “fábrica” fica parada nos primeiros dois dias de trabalho motivo pelo qual a simulação é feita para 24 dias e não 22 como sugere a ficha do projeto, sendo o período de aquecimento sendo que tem que passar por 2 armazéns para fazer tanto caixas de pranchas/rodas ou skates sendo que 2 armazéns são o equivalente à duas noites, no dia a seguir (terceiro dia), os processos começam com a produção, a classe Pressing que dá início a produção de pranchas, espera até o horário de trabalho depois passa para a classe Deck as pranchas entram no armazém e esperam até serem ativadas(a ativação é feita no início do horário de trabalho) em seguida as pranchas passam para a fila de espera do processo seguinte, de seguida é feito o pedido(“request”) pra poder ocupar o processo, espera o tempo do processo designado(em minutos) e de seguida liberta o processo em questão e sai da respetiva fila de espera podendo então passar pra o próximo processo, por fim as pranchas vão para o segundo armazém.

As rodas levam o mesmo processamento, começando na classe Foundry e passando de seguida pra classe Wheel onde a roda passa pelos diferentes processos até chegar a fase final.

Para a divisão de empacotamento dos produtos ou montagem dos skates no caso de caixas de pranchas verifica-se se a quantidade necessária é mais que a suficiente, depois de 3 em 3 dias são feitas 12 caixas de pranchas e de 2 em 2 dias 24 caixas de pranchas para o empacotamento das rodas faz-se o mesmo, verifica-se se há quantidade necessária para empacotar e de 2 em 2 dias obtemos 96 conjuntos de rodas e de 3 em 3 dias 144 conjuntos de rodas, todos os dias produz-se 240 skates.

A classe WorkTimeControl é usada para coordenar o horário de trabalho, com o state work_control, o estado fica ”working” quando nos encontramos nas 8 horas diárias de trabalho e “not_working” quando nos encontramos nas restantes 16 horas depois de o estado ser alterado para “not_working” é feita uma espera de 16 horas e em seguida é retirado todos os produtos dos armazéns faz-se a “ativação” desses mesmos produtos(pranchas e rodas).

Em termos de tempo o modelo todo aplica a distribuição Triangular que tem como parâmetros *lowerbound*, *upperbound* e *median*, para os processos a escolha do *lowerbound* e *upperbound* varia de acordo ao recurso principal, se for

humano(trabalhador) a diferença é de 10 valores caso seja uma máquina sendo a eficácia melhor a variação é de 5 valores (variação com base na média)

RESULTADO OBTIDO (seed.199 e 200)

```
(2640, 'conjunto de rodas')  
(5280, 'skates')  
(444, 'caixas de pranchas')
```

RESULTADO ESPERADO

Skates	5280
Caixas de pranchas (8 un.)	440
Conjuntos de rodas (4 un.)	2640

com este resultado realça-se o facto do modelo estar perto do sistema, certos resultados poderiam ser afetados pela distribuição usada no tempo dos processos como otimizações poderia se ter em contra os horários extras das distribuições de forma a ser usada para otimizar o funcionamento da classe WorkTimeControl que gere o funcionamento do horário de trabalho, sendo que não há proteção contra o facto de máquinas trabalharem horas extras como apresenta o resultado, obteve-se 4 caixas de pranchas à mais.

Validação

3º dia : 1344 rodas, 432 pranchas ,valores que diferem ligeiramente dos valores esperados: 1536 rodas, 396 pranchas.

4º dia : 2880 rodas, 768 pranchas, esperado: 3072 rodas 792 pranchas.

5º dia: 4224 rodas, 1200 pranchas, esperado: 4608 rodas 1188 pranchas

variações essas que podem ocorrer devido à variações estocásticas que podem afetar processos bottleneck influenciando a fluidez da fabricação