

## 5 Testes computacionais

*Por que só pensar? Por que não experimentar?*

EDWARD JENNER

Este capítulo se inicia com a descrição dos critérios e métodos utilizados para definição do conjunto de dados de teste, o qual foi utilizado para analisar o comportamento do programa YoYo em diversas situações. O capítulo é concluído com a apresentação dos resultados dos testes computacionais.

### 5.1.O programa gerador de ordens de serviço

A necessidade de desenvolvimento de um programa gerador de manobras - denominado SOGY (*Switch Order Generator for YoYo*) - surgiu durante a fase dos testes computacionais. Com o principal objetivo deste projeto é desenvolver um algoritmo para solução do PPOLM e implementá-lo, na prática, como ferramenta de apoio à decisão, desde o início havia o interesse em comparar as operações recomendadas pelo programa YoYo com as operações efetivamente executadas pelos profissionais responsáveis pelas operações do pátio em estudo.

Para comparar os dois planos, era necessário então registrar o plano gerado na prática e submeter ao programa as mesmas informações que a equipe de operação do pátio usou para elaborar o seu plano. Em seguida a idéia era comparar o custo do plano gerado pela equipe com o custo do plano gerado pelo programa.

Nos estágios iniciais do projeto, visando obter uma estimativa do retorno de investimento e justificar a viabilidade do projeto, foram feitas algumas coletas manuais de dados, simplesmente acompanhando a rotina das equipes de planejamento e execução de manobras na sala de controle de operações de pátio,

durante o intervalo de tempo de algumas horas. Nestas sessões de coletas de dados, eram registradas as ordens enviadas aos maquinistas e os tempos de execução da manobra. As mesmas informações sobre as manobras a serem executadas no intervalo de tempo amostrado eram então fornecidas ao programa YoYo e, de posse dos resultados, era possível comparar o custo das manobras executadas na prática com o custo do plano de manobras sugerido pelo programa.

Os primeiros resultados foram animadores. Os resultados obtidos naquela época com o que pode ser chamado de um protótipo do programa YoYo, conforme apresentado em Sabino (2004), foram considerados suficientes para justificar o projeto, já que sugeriam uma considerável redução de custo com a utilização da ferramenta de apoio à decisão. No entanto, a quantidade e a confiabilidade dos dados coletados com um procedimento manual são insuficientes para uma análise mais detalhada sobre a eficiência do método proposto e principalmente para o ajuste fino dos parâmetros do programa visando a sua aplicação prática. A estratégia adotada para superar estas dificuldades foi a construção de um simulador capaz de gerar um conjunto de manobras a serem executadas baseado nas características físicas e operacionais de um pátio ferroviário. As manobras geradas por este simulador foram então tomadas como modelo de referência para testes do programa YoYo.

Ocorre que o funcionamento de um pátio ferroviário é regido por vários processos que interagem nem sempre de forma previsível, o que tornou o processo de desenvolvimento do SOGY uma tarefa difícil. A estratégia adotada foi a realização de entrevistas com a equipe de planejamento para definir as variáveis que caracterizam o pátio, seguido de um estudo sobre o fluxo de operações das manobras e finalmente a validação do processo de simulação e dos resultados obtidos com profissionais experientes da área.

O pátio ferroviário submetido ao SOGY é o terminal de descarregamento fictício RRT1, cujo leiaute é apresentado na Figura 22. A linha pontilhada representa a ferrovia, de onde chegam os trens com vagões carregados e para onde vão os trens de vagões vazios depois de descarregados. As linhas contínuas representam linhas do pátio. A linha tracejada em cinza representa um trecho especial do terminal destinado exclusivamente à circulação de locomotivas de viagem. Os círculos escuros representam várias ramificações paralelas de uma

linha principal, onde podem ser estacionados vários conjuntos de vagões. Cada círculo escuro representa um conjunto padrão, constituído de uma linha principal e um total de até oito ramificações paralelas laterais de mesmo comprimento. Assim, cada círculo escuro representa uma zona do pátio. Estão representadas as zonas de recepção (RCP) e desmembramento (BKU) de trens, manutenção de locomotivas (MNS), descarregamento (UNL1, UNL2, UNL3 e UNL4), um ponto de inspeção, uma rampa de classificação (HPY) seguida de uma área de estacionamento (PRK), uma área de limpeza (CLN), uma área de manutenção de vagões (MNC) e uma área de formação de trens (MKU). Maiores detalhes sobre o leiaute do terminal RRT1 podem ser obtidos em Sabino(2004).

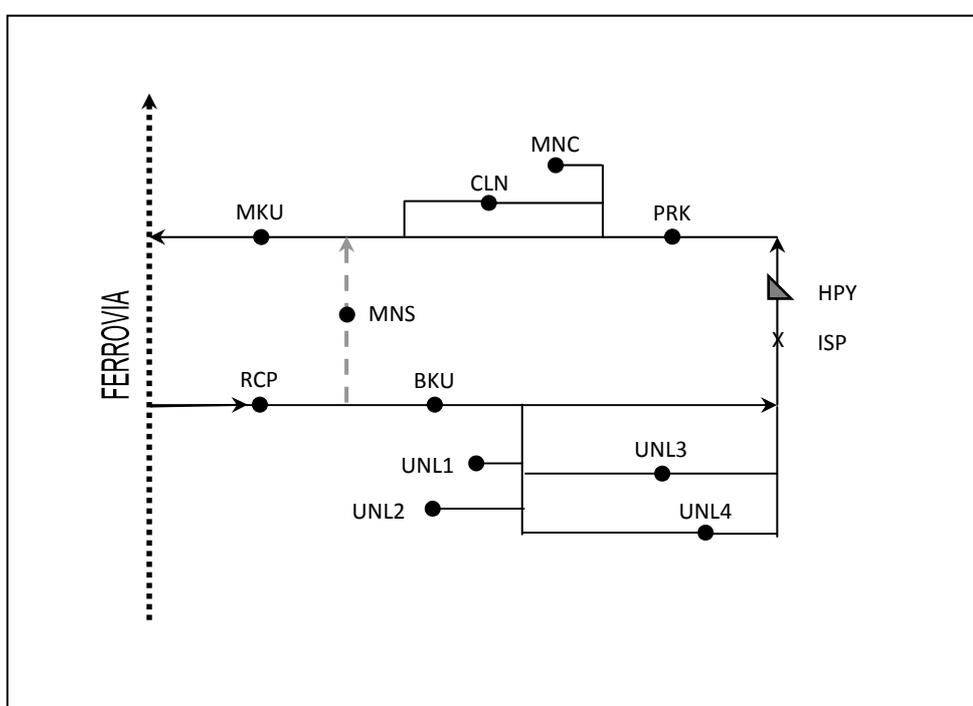


Figura 22: Pátio de descarregamento RRT1 utilizado nos testes computacionais.

A Tabela 2 mostra as variáveis de entrada para o programa SOGY, usadas para descrever as principais características do pátio fictício RRT1. Nesta tabela, as informações estão agrupadas por escopo e além da especificação do conteúdo das variáveis, estão indicados também os valores utilizados para geração dos dados de entrada para os testes computacionais deste trabalho.

Escopo	Especificação	Valor
Geral	Áreas do pátio ferroviário	RCP, BKU, UNL, PRK, CLN, MNT, MKU
Geral	Horizonte de planejamento	6 h
Geral	Peso médio de um vagão vazio	15 ton
Geral	Peso médio de um vagão carregado	50 ton
Geral	Tempo de serviço máximo permitido	2 h
Chegada	Tempo entre duas chegadas consecutivas de trens	20 min
Chegada	Vetor que contém as probabilidades de haver 1, 2, 3, 4 ou 5 blocos num trem recém chegado	{0.65, 0.15, 0.10, 0.10, 0.05}
Partida	Tempo entre duas partidas consecutivas de trens	20 min
Descarga	Tempo de serviço de uma descarga longa (assistida)	0.2 h
Descarga	Tempo médio de uma descarga de duração curta (não assistida)	15 min
Descarga	Probabilidade de uma manobra de descarga ser assistida	0.4
Descarga	Probabilidade de um conjunto de vagões carregado não descarregar durante o horizonte de planejamento dado	0.2
Inspeção	Probabilidade de um vagão requerer manutenção após a inspeção	0.05
Inspeção	Probabilidade de um conjunto de vagões requerer limpeza após a inspeção	0.01
Inspeção	Número máximo de conjuntos de vagões distintos criados após a inspeção	10
classificação	Probabilidade de um bloco localizado na área de estacionamento necessitar de manobras de classificação	0.3

Tabela 2: Parâmetros do Programa SOGY

A Figura 23 mostra o Diagrama de Fluxo de Dados do programa SOGY utilizando a notação de Yourdon & Coad (1991). O programa gera primeiramente uma base de dados relacional com a programação de chegada de trens, especificando, para cada bloco, informações como o horário de chegada do trem que contém o bloco e a sua constituição em número de vagões e tipo de carga. Esta base de dados é gerada baseado nos valores fornecidos para os parâmetros de *chegada*, conforme definido na Tabela 2.



chega ao pátio define as restrições de seqüência na execução das manobras de desmembramento.

A base de dados *switch orders* é então lida e incrementada sucessivas vezes, simulando, em ordem cronológica, as fases seguintes de descarregamento, inspeção, ida e retorno para as áreas de limpeza e de manutenção, classificação e depois formação. Como o processo é incremental, exceto quando informado explicitamente por meio de uma das variáveis da Tabela 2, cada manobra de uma etapa é pré-requisito de uma manobra correspondente gerada para a etapa seguinte. Todos os valores de probabilidade informados como parâmetros para especificar características do pátio consideram variáveis uniformemente distribuídas.

O processo *Gerar UNL* lê as manobras de desmembramento da base de dados de manobras, considera a probabilidade do conjunto de vagões movido para a área de desmembramento eventualmente não ser descarregado durante o horizonte de planejamento dado e gera as manobras correspondentes de movimentação da área de desmembramento para a área de descarregamento. O tempo de serviço das manobras de descarregamento geradas depende do parâmetro que informa a probabilidade de uma manobra ser assistida ou não e dos tempos médios de descarregamento assistido e não assistido.

Para simular a fase de inspeção são consideradas as probabilidades de um vagão requerer manutenção e de um conjunto de vagões requerer limpeza. Baseado nestas probabilidades são geradas manobras de ida e volta para inspeção e limpeza e a correspondente manobra de movimentação de vagões da área de estacionamento para a área de formação. Caso um dado conjunto ou um dado vagão não requeira limpeza nem manutenção, são geradas apenas manobras de movimentação deste vagão, ou conjunto, com coleta na área de estacionamento e entrega na área de formação.

Vale notar que o programa SOGY gera uma manobra de volta para cada manobra de ida para manutenção e que o mesmo equilíbrio se observa entre o número de conjuntos de vagões que vão e voltam da área de limpeza. Também as manobras que têm como destino a área de formação são geradas a partir da passagem de vagões pela inspeção e não de um plano de formação de trens. Estas premissas são simplificações da rotina real, que facilitam o desenvolvimento do

programa sem representar alterações significativas na similaridade das manobras geradas com as manobras usuais de um pátio real.

Com o programa SOGY, portanto, foi possível gerar o número necessário de instâncias de entrada do programa YoYo para ajuste fino dos parâmetros e comparação de métodos de solução do PPOLM.

## 5.2.

### Preparação dos experimentos e métricas utilizadas

Os objetivos dos experimentos computacionais foram:

1. Verificar se uma das duas regras de atualização do feromônio conduziria aos melhores resultados;
2. Descobrir se existiam valores recomendados para o número de formigas, o expoente associado à informação heurística  $\beta$  e o fator de persistência do feromônio  $\rho$ ;
3. Verificar se os valores recomendados para os parâmetros do algoritmo ACO mudam com a variação do tamanho da instância;
4. Verificar se o tempo médio de resposta para a obtenção de boas soluções para o problema é razoável para utilização do programa YoYo como ferramenta de apoio à decisão aplicável em situações práticas no planejamento de operações de um pátio de manobras real.

Para se obter um conjunto de entrada e dados de tamanho razoável, o programa SOGY foi usado para gerar 50 instâncias de entrada, cada uma com mais de 200 manobras a serem executadas em um horizonte de planejamento de 6 horas.

A Tabela 3 mostra os parâmetros operacionais usados nos experimentos. O horizonte de planejamento foi definido com duração igual à de um turno de trabalho do pátio real de referência deste estudo. O número de manobras a serem planejadas é a variável mais importante na determinação do tamanho da instância do problema. No caso real, observou-se uma média de 10 manobras por hora, totalizando 60 manobras em um horizonte de planejamento de 6 horas, assim, consideramos instâncias de 20 manobras, para representar uma instância de tamanho pequeno, 60 manobras para representar um valor médio para um grande

pátio de manobras (já que o caso real em estudo é o maior pátio da América Latina) e 100 manobras para representar um pátio de grandes dimensões ou com alta carga de trabalho.

A importância do custo fixo foi definida como um valor estimado, considerando o objetivo tático de um pátio real, segundo estimativas feitas em conjunto com profissionais da área de planejamento operacional de pátios. O número de locomotivas de manobras disponível foi definido bem acima da média utilizada para o pátio de referência. Isto foi feito porque o interesse maior era em determinar o número de locomotivas necessárias e não em testar a reação do algoritmo a limitações no número de locomotivas. A posição inicial das locomotivas foi definida distribuindo-as aleatoriamente pelas áreas do pátio em linhas também definidas aleatoriamente. Para determinar a capacidade de tração das locomotivas, tomou-se a capacidade dos cinco modelos de locomotivas mais utilizados no caso real e um gerador de valores aleatórios uniformemente distribuídos definiu quantas locomotivas seriam de cada um destes modelos, de modo que o total de locomotivas fosse o número dado.

Parâmetro	Notação	Valor(es)
Horizonte de planejamento (em horas)	$h$	6,0
Número de manobras a serem planejadas	$n\_orders$	20; 60 e 100
Importância relativa do custo fixo	$c_1/c_2$	0,8
Número de locomotivas disponíveis no pátio	$n\_vehicles$	40

Tabela 3: Parâmetros operacionais utilizados nos experimentos computacionais

Os valores dos parâmetros do algoritmo ACO utilizados nos experimentos são mostrados na Tabela 4. O número inicial de formigas para ambas as colônias no algoritmo CompetAnts foi 40, então decidiu-se tomar três valores maiores que este para investigar até onde valia a pena investir em processamento adicional (i.e. aumentando o número de formigas) visando a obtenção de soluções melhores. Esta estratégia também serviu para validar se o número de formigas recomendado em Dorigo & Stützle (2004) para bom desempenho na solução do Problema do Caixeiro Viajante era bom também para o PPOLM. Vale notar que a orientação de que o número de formigas deve ser igual ao número de cidades do Problema do Caixeiro Viajante deve ser convertida na especificação do número de formigas

igual ao dobro do número de manobras a serem executadas no PPOLM, já que cada manobra está associada a dois pontos de parada na rota de uma locomotiva.

O expoente do feromônio, o valor inicial do feromônio e o número de formigas que depositam feromônio foram definidos de acordo com as recomendações de parâmetros proposta em Dorigo & Stützle (2004) para bom desempenho do algoritmo *Ant System* na solução do Problema do Caixeiro Viajante. A importância do feromônio da própria colônia em relação ao feromônio da outra colônia foi definida com o mesmo valor usado em Reimann (2002), ou seja, de modo que as formigas dêem a mesma importância ao feromônio de ambas as colônias.

Devido à simplicidade de implementação e o menor tempo de execução esperado, optou-se por utilizar o custo variável estático no cálculo do valor da função objetivo. Diante da impossibilidade de desenvolver um sistema eficiente para estimar, com razoável precisão, os tempos de deslocamento e espera durante a realização das manobras, os eventuais conflitos de alocação de linhas pelos comboios durante a execução do plano de manobras proposto foram deixados para tratamento em tempo real pelas equipes de operação do pátio.

Parâmetro	Notação	Valor(es)
Número inicial de formigas (para ambas colônias)	$m$	40; 60; 100 e 200
Expoente do feromônio	$\alpha$	1
Expoente heurístico	$\beta$	1; 3; 5 e 10
Fator de evaporação de feromônio	$\rho$	0,02; 0,10; 0,50; 0,9 e 0,98
Valor inicial do feromônio	$\tau_0$	0,1
Importância do feromônio da própria população	$\chi$	0,5
Número de formigas que depositam feromônio (apenas para a regra RNK)	$w$	6

Tabela 4: Parâmetros do algoritmo ACO utilizados nos experimentos computacionais

As versões RNK e CME do algoritmo foram executadas uma única vez (Birattari, 2005) para 50 listas diferentes de manobras geradas pelo programa SOGY. Para cada instância de manobras foram testadas todas as combinações possíveis dos valores de parâmetros mencionados nas tabelas 3 e 4. Esta estratégia de testes resultou em um conjunto de 240 combinações de parâmetros a serem

submetidas para cada algoritmo, produzindo  $240 \times 50 \times 2 = 24000$  resultados a serem avaliados.

O programa termina depois de concluir a trigésima iteração do algoritmo ACO. Este era o único critério de parada e, ao final, as seguintes informações eram registradas para análise futura:

- O valor da solução, ou seja, o melhor valor encontrado para a função objetivo depois de 30 iterações;
- O tempo de CPU consumido até encontrar a melhor solução. Vale notar que este não é, necessariamente, o tempo total de CPU consumido após as 30 iterações, já que, na maioria das vezes, a melhor solução é encontrada antes da última iteração.
- O número de iterações até o ponto em que a melhor solução foi encontrada. Este valor foi registrado apenas para tornar possível estimar o impacto da redução do número de iterações necessárias na qualidade da solução.

Os dados coletados foram sumariados utilizando uma medida de tendência central ao invés do melhor resultado encontrado (Birattari & Dorigo, 2007). Foi usada análise de variância (ANOVA) com nível de significância de 5% para comparar os resultados e avaliar se existia evidência de diferença entre as médias das populações. Os conjuntos de dados com valores do número de manobras diferentes não foram consolidados porque eles representam as diferentes cargas de trabalho submetidas ao pátio e, portanto, devem ser analisados separadamente.

Ambos os algoritmos foram implementados em ANSI C e o código fonte foi compilado com gcc versão 3.4.6. Todos os testes foram executados em ambiente Red Hat Enterprise Linux, CentOS release 4.5 instalados em servidores AMD opteron™ 244 com clock de 1,75 GHz, 1 Mb de memória L2 cache e 2 Gb de memória RAM.

### **5.3.Resultados**

A primeira análise feita foi uma comparação direta entre a qualidade da solução obtida com a implementação das regras CME e RNK. A Figura 24 e a Tabela 5 apresentam a média e o desvio padrão do valor da solução, considerando

todos os testes feitos para conjunto de dados de entrada contendo 20, 60 e 100 manobras. O histograma da Figura 24 mostra o resultado lado a lado e os respectivos valores com precisão de 4 casas decimais são mostrados na Tabela 5. Esta primeira análise sugere que não há diferença significativa na qualidade da solução para nenhum dos valores de  $n\_orders$ . A análise de variância suporta esta hipótese intuitiva, já que a mesma não produz evidências de diferença estatisticamente significativa entre os resultados obtidos para os algoritmos que utilizam as regras CME e RNK.

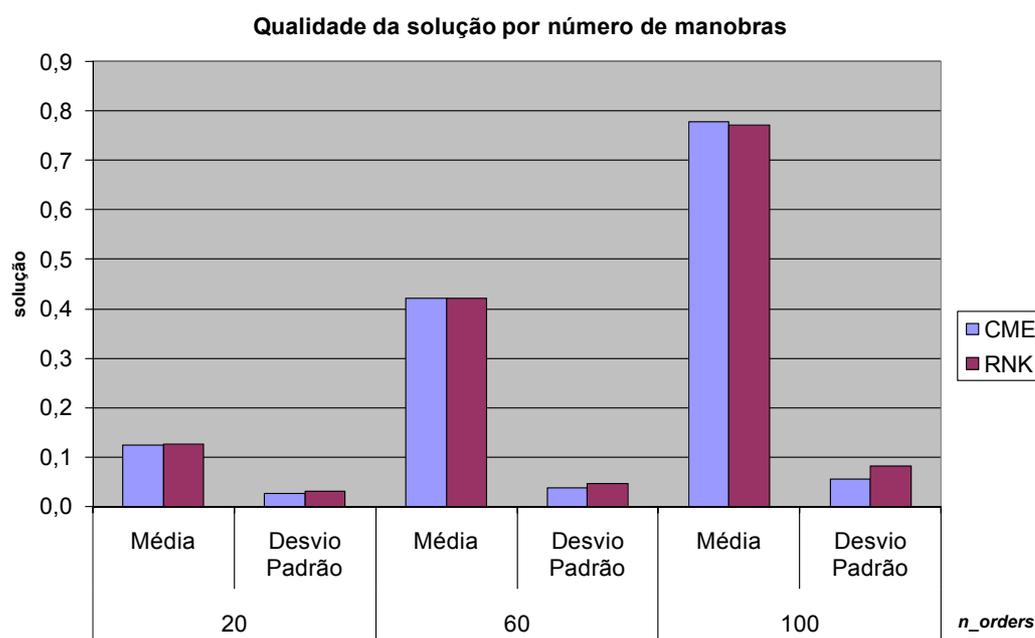


Figura 24: Comparação da qualidade da solução para as regras CME e RNK

A primeira conclusão foi, então, que com base no sumário de todos os resultados obtidos, agrupados por número de manobras e sem considerar os parâmetros utilizados em cada caso, não é possível concluir que um dos dois algoritmos testados apresenta melhor qualidade que o outro.

		Algoritmo	
n_orders	Solução	CME	RNK
20	Média	0,1246	0,1270
	Desvio Padrão	0,0274	0,0306
60	Média	0,4213	0,4212
	Desvio Padrão	0,0366	0,0466
100	Média	0,7789	0,7719
	Desvio Padrão	0,0545	0,0828

Tabela 5: Comparação da qualidade da solução para as regras CME e RNK

Também o tempo de CPU, conforme apresentado na Tabela 6, apresentou um desvio padrão relativamente grande para os três valores de número de manobras testados, tornando, portanto, impossível identificar qualquer diferença estatisticamente significativa.

A conclusão, baseada nos resultados gerais agrupados por número de manobras, é que nenhum dos dois algoritmos testados produz resultados melhores que o outro no que se refere a qualidade e custo das soluções.

		Algoritmo	
n_orders	Tempo de CPU (s)	CME	RNK
20	Média	1,29	1,17
	Desvio Padrão	2,55	2,35
60	Média	29,89	23,08
	Desvio Padrão	33,81	33,94
100	Média	79,71	71,62
	Desvio Padrão	78,43	100,40

Tabela 6: Tempo de CPU, em segundos, das soluções CME e RNK

O próximo passo foi verificar a influência dos parâmetros ACO na qualidade da solução, sendo de particular interesse a identificação do valor recomendado para os três parâmetros ACO que foram modificados: o número inicial de formigas das duas colônias  $m$ , o expoente da atratividade  $\beta$  e o fator de evaporação de feromônio  $\rho$ .

O histograma da Figura 25 compara a média das soluções para cada valor de  $m \in \{40, 60, 100, 200\}$ . As barras estão agrupadas de acordo com o número de manobras do conjunto de dados de entrada.

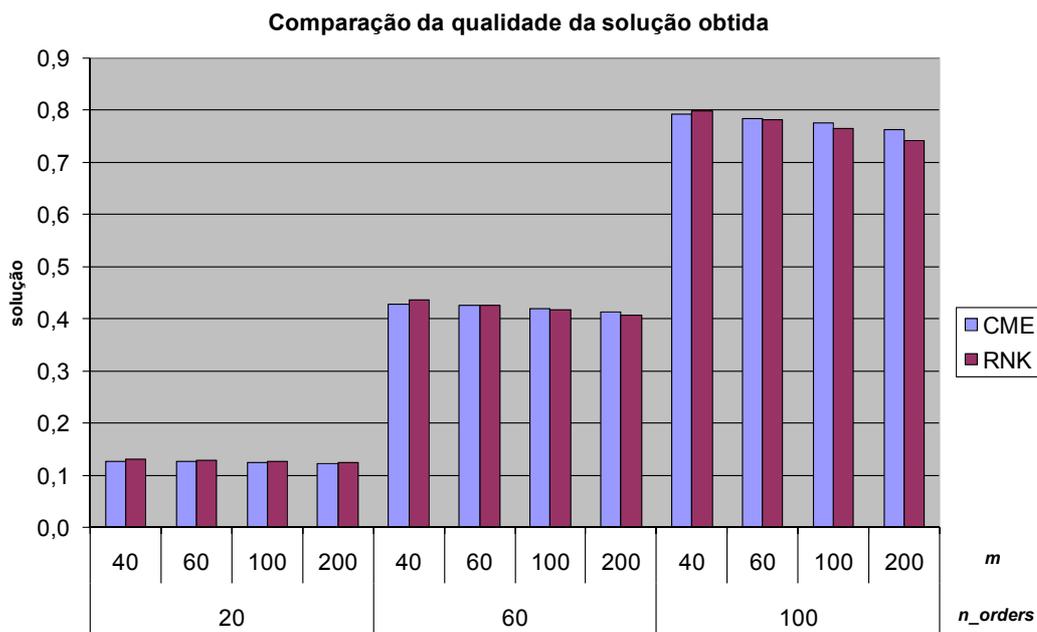


Figura 25: Comparação das soluções CME e RNK variando  $n\_orders$  e  $m$

Percebe-se que não há diferença significativa na qualidade da solução para  $n\_orders=20$ . Para  $n\_orders=60$  e  $n\_orders=100$ , independente do algoritmo, a solução melhora à medida que o valor de  $m$  aumenta. Além disso, o algoritmo baseado na regra RNK produz melhores resultados que o que utiliza a regra CME para  $m = 100$  e  $m = 200$ . Todas estas conclusões são suportadas pelo teste estatístico ANOVA. Por exemplo, na Tabela 7 o sumário contendo o resultado da análise de variância simples das amostras com  $m = 200$  e  $n\_orders=60$ , considerando um nível de significância de 0,05 mostra que existe uma diferença estatisticamente significativa entre os valores das soluções quando se compara as regras CME e RNK.

Anova: fator único

**RESUMO**

Grupo	Contagem	Soma	Média	Variância
CME	1000	413,1341	0,413134	0,001293
RNK	1000	406,31	0,40631	0,001678

**ANOVA**

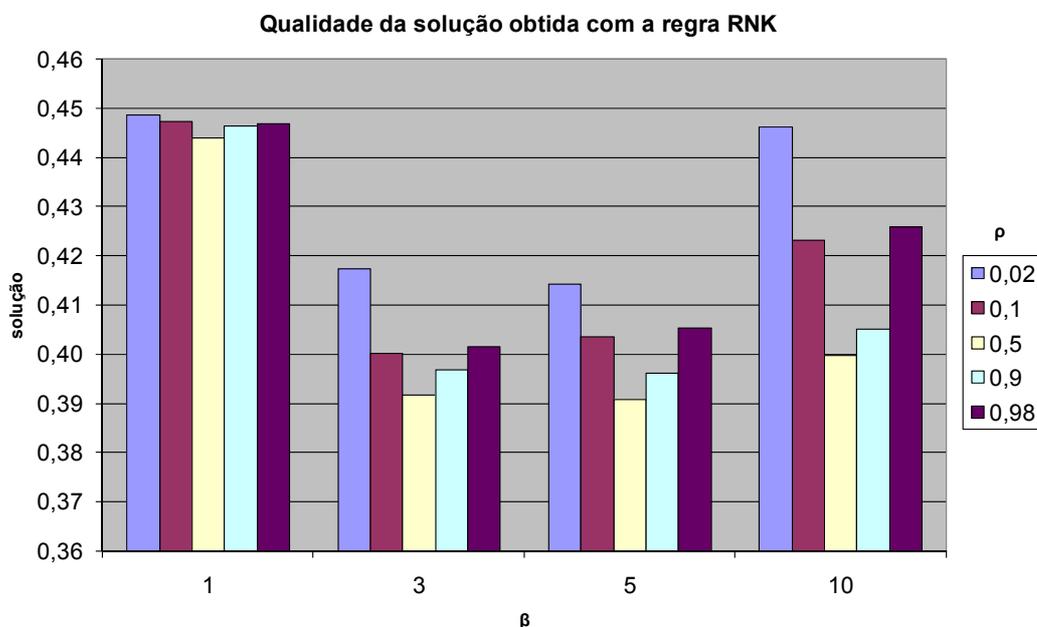
Fonte da variação	SQ	gl	MQ	F	valor-P	F crítico
Entre grupos	0,023284	1	0,023284	15,67681	7,78E-05	3,846117
Dentro dos grupos	2,967564	1998	0,001485			
Total	2,990848	1999				

Tabela 7: Resultados de ANOVA com fator único para  $m = 200$  e  $n\_orders=60$

O fato de valores maiores de  $m$  levarem a resultados melhores é uma consequência do número adicional de soluções geradas quando se aumenta o valor de  $m$ .

Quanto aos valores de  $\rho$  e  $\beta$ , chegou-se às seguintes conclusões após analisar a média das soluções encontradas para combinações de  $m$ ,  $\rho$  e  $\beta$  para cada valor possível de  $n\_orders$ :

- Para o algoritmo baseado na regra RNK,  $\rho=0,5$  produz as melhores soluções, especialmente para valores maiores de  $m$  e independente do valor de  $\beta$ . Isto é válido para qualquer valor de  $n\_orders$  e a melhor combinação é  $\rho=0,5$  e  $\beta=5$ ;
- Para o algoritmo CME,  $\rho=0,98$  e  $\beta=3$  é a melhor combinação considerando fixos  $n\_orders=60$  e  $n\_orders=100$ . Os melhores valores de  $\rho$  e  $\beta$  nem sempre são os mesmos. Por exemplo, para  $n\_orders=20$  a melhor combinação é  $\rho=0,98$  e  $\beta=1$ ;
- A regra RNK com sua melhor combinação  $\rho=0,5$  e  $\beta=5$  produz uma solução melhor que qualquer combinação de parâmetros testados com a regra CME para  $n\_orders=60$  e  $n\_orders=100$ .



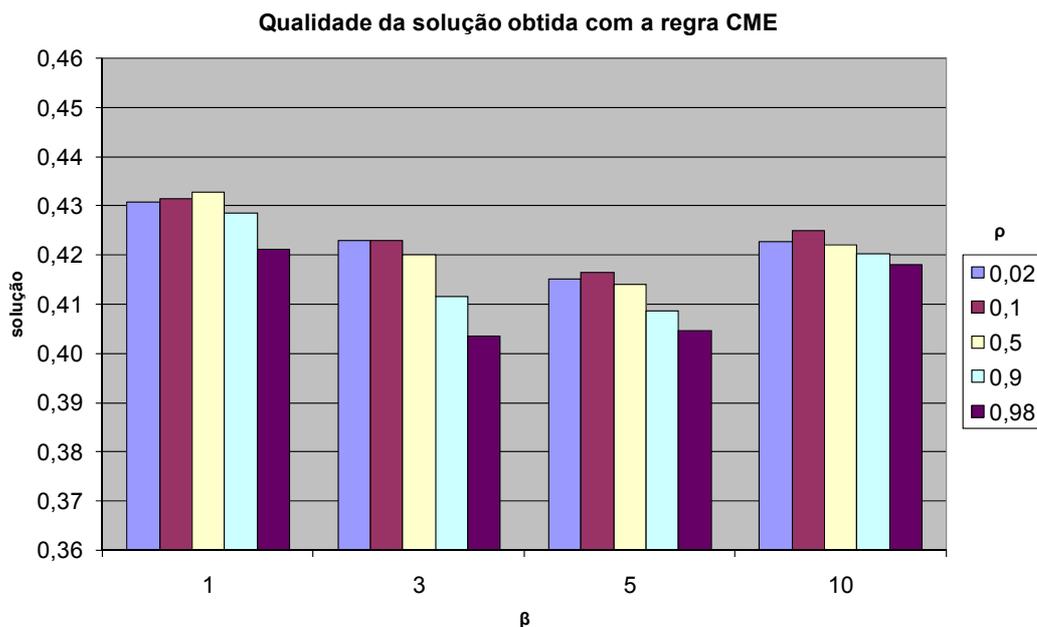


Figura 26: Comparação das regras RNK e CME variando  $\rho$  e  $\beta$

Esta última conclusão é ilustrada com os dois histogramas da Figura 26 que apresentam no eixo y a média das soluções obtidas utilizando as regras CME e RNK, mantendo fixos  $m=100$  e  $n\_orders=60$ . Cada cor de barra no eixo x representa um valor de  $\rho$  e cada conjunto agrupado de cinco barras representa um valor de  $\beta$ , assim, para cada um dos quatro valores de  $\beta$  testados são mostrados os valores correspondentes de  $\rho$  dispostos lado a lado. As barras estão dispostas com os valores de  $\beta$  e  $\rho$  classificados em ordem crescente da esquerda para a direita.

Como os dois gráficos utilizam a mesma escala para o eixo y é fácil observar que os melhores resultados são obtidos com o algoritmo RNK com parâmetros  $\rho=0,5$  e  $\beta=5$ .

Uma análise mais detalhada das características das regras CME e RNK indica uma possível razão para o melhor desempenho da regra RNK: Como a lógica da regra CME não repassa as informações sobre o desempenho das iterações anteriores para as iterações seguintes, a regra RNK explora mais agressivamente a melhor solução obtida até então e portanto é guiada mais diretamente para as melhores soluções.

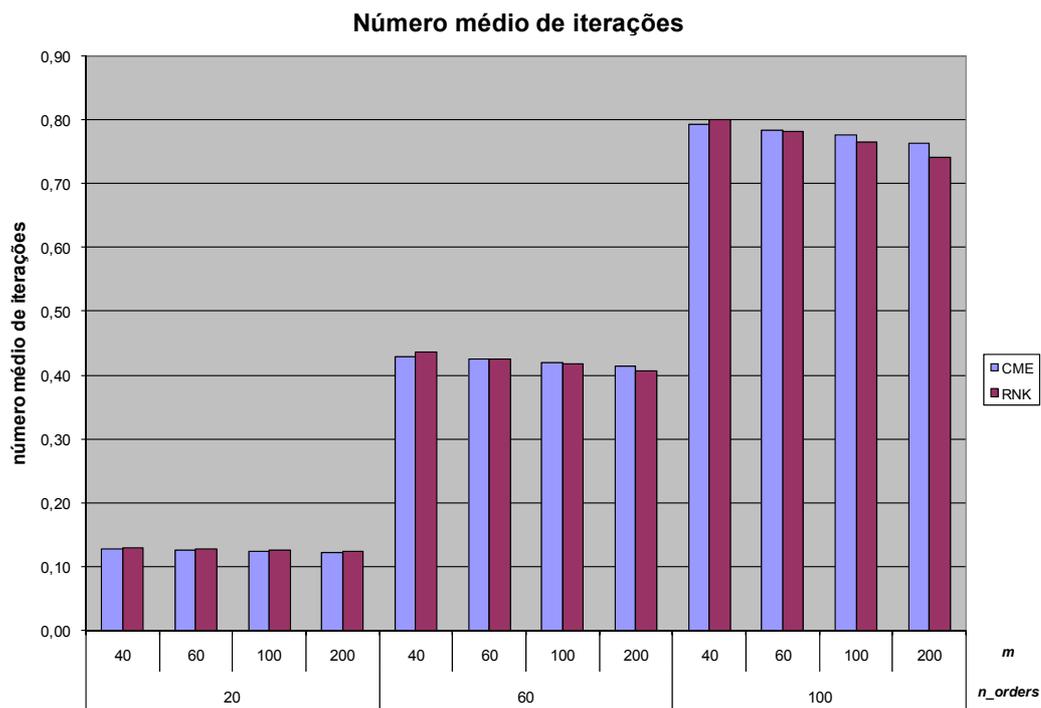


Figura 27: Número médio de iterações para CME e RNK variando  $n\_orders$  e  $m$

A Figura 27 mostra, para os valores de  $m$  e  $n\_orders$  testados, a média do número de iterações necessárias até encontrar a melhor solução, considerando  $\rho=0,5$  e  $\beta=5$ . As barras correspondentes aos resultados obtidos com as regras CME e RNK estão emparelhadas para cada valor de  $n\_orders$  e agrupadas por valor de  $m$ . É fácil observar que o número de iterações necessárias para encontrar a melhor solução aumenta com o valor de  $n\_orders$ . Como o desvio padrão do número de iterações necessárias até encontrar a melhor solução é relativamente alto em todos os casos testados, não foi possível obter diferença estatisticamente significativa entre o número de iterações para os valores testados de  $n\_orders$  e  $m$ .

De uma forma geral, o número de iterações e, portanto, o tempo de CPU necessário para se chegar à solução do problema foram tão dispersos que ficou impossível de se chegar a alguma conclusão sobre a comparação do custo computacional para obtenção das soluções. O gráfico da Figura 28 dá uma idéia de quão dispersos estão os valores de tempo de CPU necessário considerando todas as combinações dos parâmetros testados mantendo fixo  $n\_orders=60$ . No gráfico do tipo *boxplot* as extremidades superior e inferior das barras verticais indicam a posição do primeiro e do terceiro quartil de dados respectivamente. As extensões pontilhadas vão até o valor que corresponde a 1,5 vezes a faixa

interquartil indicada pela barra. Os valores atípicos, ou seja, que apresentam um grande afastamento dos restantes, são indicados como pequenos círculos.

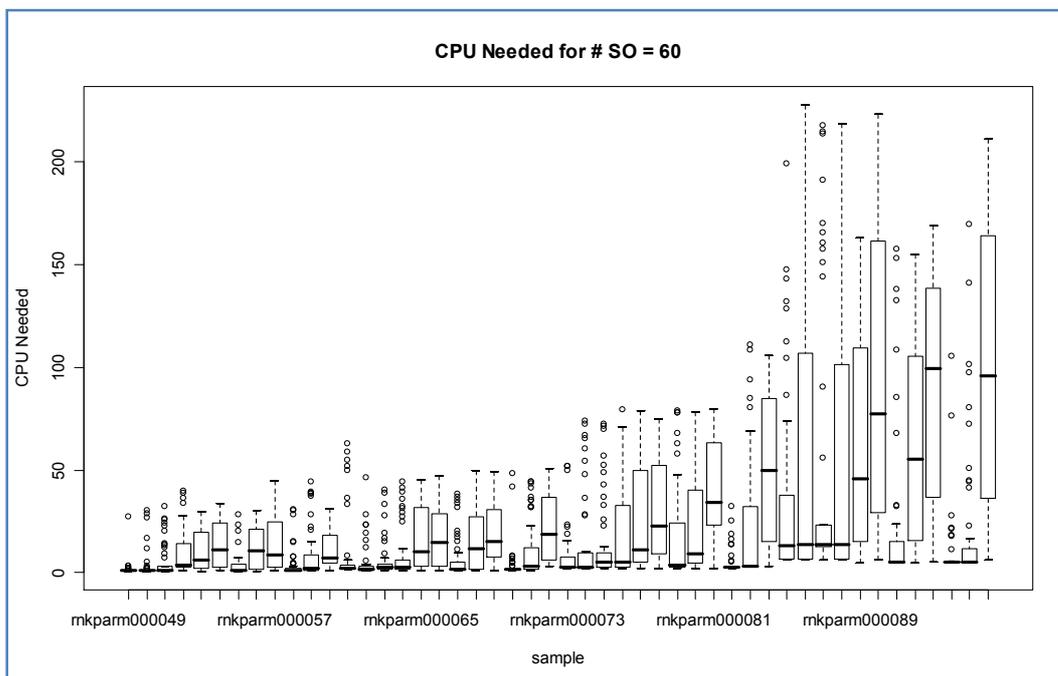


Figura 28: Boxplot do tempo de CPU para todas as combinações de parâmetros considerando  $n\_orders=60$

Para verificar a possibilidade de utilização do algoritmo proposto para resolver instâncias reais, foi avaliado o tempo de resposta das 50 execuções com o algoritmo RNK utilizando a melhor combinação de parâmetros e considerando  $n\_orders=60$ , que é o tamanho de instância mais realista testada. O valor médio deste tempo de resposta foi 82 segundos, com um desvio padrão de 0,5 segundos. Este resultado corresponde às expectativas para aplicação prática, já que neste tempo é possível refazer o planejamento de manobras mais de 100 vezes durante um horizonte de planejamento padrão de 6 horas de duração. O número médio de locomotivas de manobras necessárias para a solução destas instâncias de tamanho real foi 4, o mínimo foi 3 e o máximo foi 7. Estes valores estão abaixo do esperado para um pátio com esta carga de trabalho, segundo opinião de profissionais da área. Estes resultados sugerem que a implementação do algoritmo RNK para suporte à decisão de planejamento operacional de pátios pode reduzir, inclusive, o número de locomotivas de manobras necessárias para a operação do pátio.