

UMA METODOLOGIA PARA SOLUÇÃO DE PROBLEMAS DE GRANDE ESCALA DE VMI – *VENDOR MANAGED INVENTORY*

Dr. Nelson Standerski

Resumo

Este trabalho propõe uma metodologia baseada em técnicas metaheurísticas paralelas e em computação de alto desempenho para a solução de problemas de reabastecimento ótimo de estoques - VMI (*Vendor Managed Inventory*).

O modelamento admite entregas diretas rodoviárias a uma grande base de clientes. Cada cliente é caracterizado pela sua localização geográfica, capacidade máxima de armazenamento, taxa de consumo e nível inicial de estoque. O modelo utiliza a formulação clássica determinística e linear de reposição ótima de estoques e

A solução do modelo é feita utilizando-se uma infra-estrutura computacional distribuída que utiliza o tempo ocioso de redes locais de PCs (noites e fins-de-semana). A infra-estrutura computacional é baseada em linguagem java e simula a operação de um supercomputador virtual de alto desempenho.

Experimentos realizados com um modelo com grande número de pontos de reabastecimento (4.000) mostraram que o tempo de cálculo do modelo se reduz consideravelmente com a utilização de processamento paralelo.

A metodologia proposta permite ampliar de dezenas para milhares o número de clientes que podem ser analisados simultaneamente em VMI. A solução proposta apresenta grande vantagem econômica pois utiliza apenas recursos computacionais ociosos.

Palavras-chave: **logística, VMI, algoritmos genéticos paralelos.**

XXIII Seminário de Logística Internacional - Associação Brasileira de Metalurgia e Materiais – Belo Horizonte, 16 a 18 de Junho de 2004.

Nelson Standerski é Engenheiro e MSc. pela Escola Politécnica da USP, e PhD pela Universidade Técnica de Berlim, Alemanha. Foi pesquisador do IPT Instituto de Pesquisas Tecnológicas e gerente de engenharia na Siemens. Desde 1995 atua como consultor na Paperless P&D Ltda (www.paperless.com.br).
E-mail: nelson@paperless.com.br

Este trabalho tem o apoio da **FAPESP** – Fundação de Amparo a Pesquisa do Estado de São Paulo - Programa PIPE – contrato número 01/03196-6.

1. Introdução

Otimização logística é uma ferramenta de grande importância para redução de custos em indústrias cuja cadeia de suprimentos utiliza matérias-primas e produtos de menor valor específico, como é o caso da indústria siderúrgica. Em indústrias desse tipo, os custos com transporte podem chegar a quase 50% do custo final do produto. O gerenciamento de estoques nesse tipo de cadeia de suprimentos preocupa-se principalmente em garantir a disponibilidade ininterrupta dos insumos, de forma a não afetar a produção de produtos de maior valor específico.

VMI – *Vendor Managed Inventory* – é uma técnica de planejamento logístico que determina os tempos ótimos de reabastecimento de estoques. As vantagens principais dessa técnica são: garantir que nunca haja falta de estoques e reduzir custos de transporte.

Problemas de otimização de VMI são usualmente tratados utilizando técnicas exatas de busca, como MIP (*mixed-integer programming*) e seus derivados (*branch-and-bound*, *branch-and-cut* e *branch-and-price*). Em problemas de pequeno porte, métodos exatos produzem melhores soluções tanto em termos de tempo de cálculo como de proximidade com a solução ótima. Essas técnicas têm sido empregadas com muito sucesso em problemas de VMI [1] [2].

Porém, quando aplicadas a modelos de grande escala (grande base de clientes), técnicas exatas de busca não mostram a mesma eficiência, resultando em elevados tempos de processamento. Muitas vezes nem mesmo uma única solução viável para o problema é encontrada. A figura 1 [3] abaixo mostra os resultados de testes realizados para solução de problemas de otimização logística com um *solver* paralelo comercial. Observa-se na figura 1 que a medida em que se aumenta o tamanho do problema (mais clientes e/ou maior número de visitas), maior a dificuldade em se encontrar uma

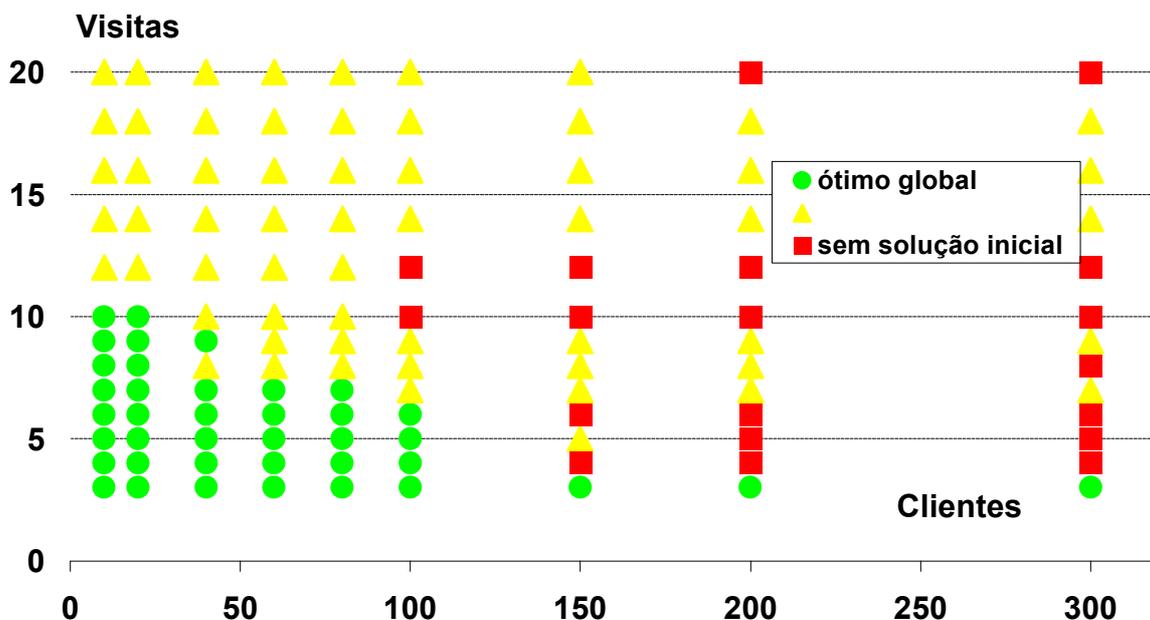


Figura 1 – Experimentos com programação inteiro-mista.

problemas reais de logística. É comum encontrar-se na prática funções logísticas não descritas analiticamente.

Recentemente tem-se aplicado com sucesso a técnica de algoritmos metaheurísticos paralelos a problemas de otimização de grande escala, por exemplo, em manufatura discreta e em telecomunicações [4].

Algoritmos metaheurísticas são intrinsecamente paralelizáveis, e sua aplicação em ambiente computacional paralelo podem reduzir significativamente o tempo total de processamento dos modelos.

O objetivo principal deste estudo é o de apresentar uma metodologia para solução de problemas de grande escala de VMI. Os principais componentes desta metodologia estão apresentados esquematicamente na tabela 1 abaixo, e de forma detalhada nos

Modelamento	Algoritmos	Software	Hardware
<i>VMI</i>	<i>Metaheurísticas</i>	<i>Programação paralela</i>	<i>Computação alto desempenho</i>
Modelo linear e determinístico de estoques	Algoritmos genéticos	Tecnologia Java	Cluster de PCs

Tabela 1 – Esquema geral da metodologia proposta.

2. Modelo de VMI

2.1. Definição do Problema

VMI – *Vendor Managed Inventory* – é uma prática de operação logística na qual os fornecedores são responsáveis pelo gerenciamento dos estoques de seus clientes, e devem garantir que nunca haja desabastecimento do produto. A vantagem dessa prática é que o fornecedor pode programar as suas entregas com antecipação, permitindo utilizar mais racionalmente a sua frota. Essa prática tem-se desenvolvido bastante ultimamente devido ao barateamento dos custos de monitoramento remoto de estoques.

Na operação convencional (sem VMI), o cliente coloca seus pedidos periodicamente, normalmente no início de cada semana ou mês, ou quando verifica que os níveis de estoque já estão muito baixos ou mesmo nulos.

O resultado é que o fornecedor recebe em determinados dias da semana ou do mês uma grande quantidade de pedidos a serem atendidos simultaneamente, muitos deles com urgência. A operação convencional é uma operação crítica, em que normalmente ocorre desabastecimento no cliente e superdimensionamento no tamanho da frota de distribuição do fornecedor.

Com VMI os clientes não fazem mais pedidos. Seus estoques são medidos

remotamente por telemetria e a previsão de consumo é feita por meio de estatísticas. VMI oferece a vantagem para o cliente de sempre dispor do produto, e para o fornecedor de programar com maior antecipação suas entregas.

Um caso particular de VMI, mas de grande interesse prático, é aquele em que a entrega é feita a um cliente de cada vez, utilizando um veículo com carga cheia, que após o reabastecimento retorna a sua base. Esse será o caso estudado neste trabalho.

2.2. Modelo de Estoques

O modelo de estoques utilizado, descrito detalhadamente em [5], admite que as taxas de consumo do produto no cliente são uniformemente distribuídas ao longo do tempo. Na realidade, entretanto, as taxas de consumo podem variar bastante, por exemplo, devido a um aumento pontual excessivo de consumo. Por isso é conveniente rodar o modelo diariamente, incorporando em tempo real os níveis atuais de estoque em cada cliente, e reduzindo a importância da previsão estatística no modelo.

uniformidade da taxa de consumo, resulta o gráfico de variação do nível de estoque em cada cliente apresentado na figura 2.

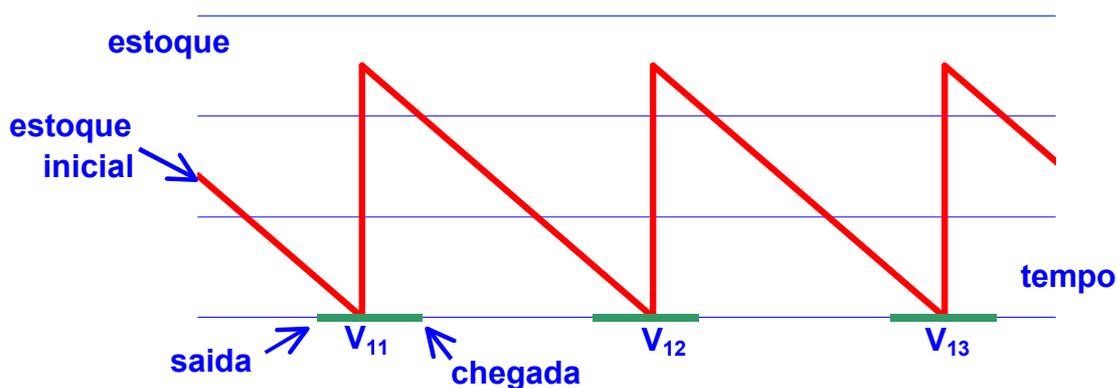


Figura 2 – Tempo de reabastecimento ótimo.

O tempo ótimo V_{ij} da visita j (estoque zero) para um determinado cliente i é dado por:

$$V_{ij} = I_i^0 / u_i, \quad \text{para } j = 1$$

$$V_{ij} = Q / u_i + V_{ij-1}, \quad \text{para } j > 1$$

onde V_{ij} é o tempo da visita j no cliente i , Q a capacidade de transporte dos veículos, u_i a taxa de consumo e I_i^0 o estoque inicial no cliente i .

A figura 2 anterior também mostra o intervalo de tempo em que o veículo está em viagem. No caso de dois locais de reabastecimento, pode ocorrer que seus tempos de viagem se sobreponham, e portanto dois veículos sejam necessários para atender os clientes no tempo de reabastecimento ótimo V_{ij} .

Do ponto de vista logístico seria muito interessante reduzir o número de veículos necessários, ou seja, com apenas um veículo fosse possível atender às entregas dos dois locais, sem ocasionar desabastecimento em nenhum local.

Como seria possível fazer isso ?

Uma possibilidade é antecipar convenientemente as entregas sem que haja

sobreposição no uso dos veículos.

A antecipação da entrega está limitada superiormente pela capacidade máxima de armazenamento, ou seja, não é possível reabastecer um cliente quando seus níveis de estoque ainda estão muito altos, próximos do limite máximo. A postergação da entrega também não é possível, pois implicaria desabastecimento no cliente.

Definindo-se a variável a_{ij}

V_{ij} , o tempo real de entrega resulta:

$$t_{ij} = V_{ij} - a_{ij}$$

sendo que $a_{ij} \geq 0$ para que não haja desabastecimento, e $a_{ij} \leq (C_i - Q) / u_i$ para que a capacidade máxima de armazenamento C_i não seja ultrapassada.

Numa situação de apenas dois clientes, a solução é encontrada por simples inspeção. Porém no caso de muitos clientes, por exemplo 4.000, as possibilidades de combinação são praticamente infinitas.

2.3. Formulação do Problema de Reabastecimento Ótimo

O problema anterior pode ser tratado como um problema de otimização.

Uma possível estratégia de otimização é aceitar inicialmente uma frota grande suficiente para atender a todos os clientes sem que haja desabastecimento, e em seguida buscar soluções em que a sobreposição do uso dos veículos seja minimizada. Dessa forma minimizam-se os custos fixos da frota. Os custos operacionais dos veículos (combustível, lubrificantes etc) são os mesmos para todas as so os veículos operam em entrega direta. Esta estratégia de solução permite que pelo

O tamanho da frota varia ao longo do tempo em função da programação de entregas. A variação da frota com o tempo está exemplificada na figura 3 abaixo.

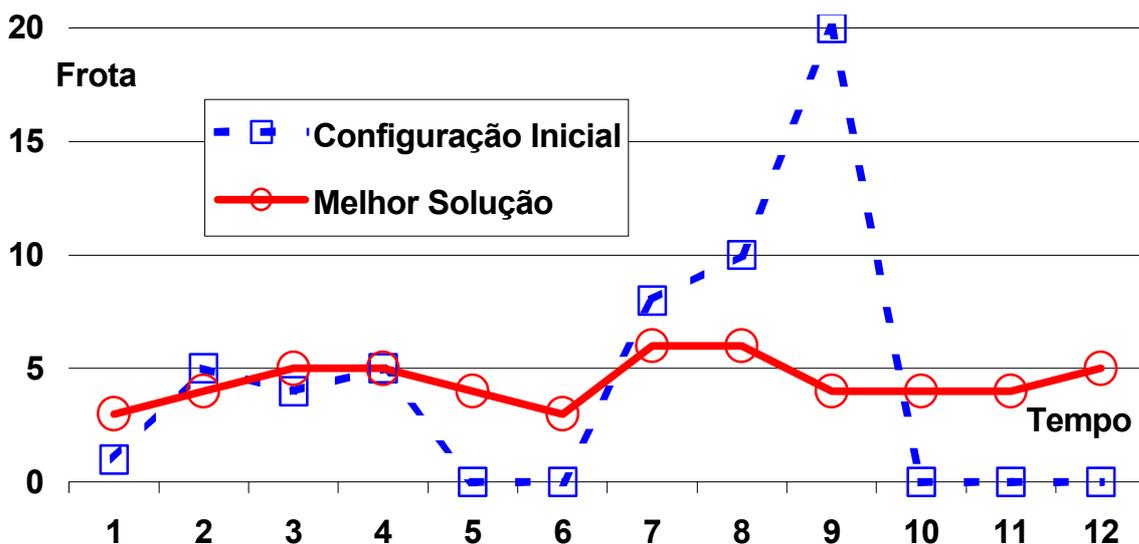


Figura 3 – Variabilidade do tamanho da frota.

As duas curvas hipotéticas apresentadas na figura 3 atendem em princípio às restrições do problema. Mas a solução com menor variação de frota ao longo do tempo

(linha contínua) é mais interessante do ponto de vista logístico, pois tem menor custo fixo de frota. Por isso adota-se a minimização do erro quadrático médio do tamanho da frota ao longo do tempo como função objetivo para o problema de otimização de VMI. entável explicitamente por meio de fórmulas analíticas.

A função objetivo pode ainda incorporar o custo adicional de capital imobilizado dos estoques nos clientes, que deverá ocorrer devido à entrega antecipada do produto. O modelo pode neste caso verificar se em termos de custo é melhor uma frota maior e menores estoques nos clientes, ou o contrário.

Outras hipóteses simplificadoras são admitidas: modelos lineares de cálculo de custos de transporte e estoque, único produto, tempos de carga e descarga nulos, sem janelas de tempo e de horas de *rush*, único local de carregamento. Admite-se também que o tempo de viagem é linearmente proporcional à distância e independente do veículo e do período em que é realizado. Essas hipóteses, no entanto, não precisariam ter sido impostas. Foram admitidas para facilitar o desenvolvimento e entendimento do modelo.

Com base nessas informações, o modelo de VMI é formulado como um problema de otimização composto por uma função objetivo e uma série de restrições operacionais (velocidades, capacidade dos veículos, distâncias, estoques mínimo e máximo etc).

3. Algoritmos Genéticos

Algoritmos Genéticos são métodos heurísticos de busca inspirados nos mecanismos de evolução de populações de seres vivos. Esses algoritmos seguem a teoria das espécies proposta por Charles Darwin. Segundo essa teoria, quanto melhor um indivíduo se adaptar ao seu meio ambiente, maiores as suas chances de sobrevivência. Todos os organismos vivos consistem de células. Em cada célula há um mesmo conjunto de cromossomos por sua vez compostos por genes. A seqüência desses genes define todas as características do organismo, inclusive a sua maior ou menor adaptabilidade ao meio-ambiente.

Durante a reprodução ocorre uma mistura entre os genes dos pais e um cromossomo com uma nova seqüência de genes é criado. Como pela teoria de evolução somente os pais com melhor adaptabilidade sobrevivem, espera-se que depois de muitas gerações (evolução), a população esteja bem adaptada ao meio ambiente.

Aplicando esse conceito ao problema de VMI estudado, soluções (indivíduos) com uma frota de veículos menor (melhor adaptabilidade) devem ser sempre obtidas ao se combinar soluções (pais) por muitas iterações (gerações).

O processo de evolução no caso do problema de VMI inicia-se com a geração aleatória das datas de reabastecimento de estoques (genes) para um determinado número (população) de soluções (cromossomos). As soluções são então classificadas segundo o valor de sua função de aptidão (por exemplo tamanho da frota). Um -se as melhores soluções da população anterior e criando-se novas soluções a partir da recombinação ou mutação de genes de soluções da população anterior. O processo de evolução geracional prossegue até que seja atingido algum critério de parada, p.ex., número de gerações, tempo de processamento, invariabilidade dos indivíduos de uma população etc.

No caso do modelo de VMI de grande escala, a representação cromossômica proposta por Kemenade e Kok [6] é particularmente eficaz pois utiliza variáveis reais como genes. A tabela 2 abaixo mostra a representação cromossômica utilizada. Trata-se de

uma matriz bidimensional onde a_{ij} são as variáveis reais do problema.

	1	2	3	...	N clientes
1	a_{11}	a_{12}	a_{13}	...	a_{1N}
...					
M visitas	a_{M1}	a_{M2}	a_{M3}	...	a_{MN}

Tabela 2. Estrutura cromossômica.

A criação de novas gerações no modelo de VMI é feita por meio de mutação aleatória no valor dos genes dos indivíduos da geração anterior. O intervalo de sorteio do novo gene varia uniformemente com o valor da função de aptidão de seu cromossomo. A melhor solução terá intervalo de sorteio nulo e permanecerá inalterada para a nova solução. A pior solução terá intervalo de sorteio máximo permitido pelas restrições das variáveis. Dessa forma, sempre será feita uma busca local e global em torno da melhor solução da geração anterior.

Algoritmos genéticos oferecem boas possibilidades de paralelização [4]. Por exemplo, dividindo-se a população em subgrupos e executando cada subproblema em processadores diferentes. Depois de determinado número de gerações, as subpopulações são comparadas entre si (operador genético de migração), e os melhores indivíduos (soluções) são intercambiados entre os nós processadores.

A busca prossegue independentemente em cada processador e, após um determinado número de gerações, quando um novo intercâmbio de indivíduos é realizado. Esse processo é repetido até que um critério de parada final seja atingido.

4. Supercomputador Virtual - JavaWulf

A solução clássica de problemas paralelos é feita com estações de trabalho multiprocessadas de alto desempenho ou com agregados de computadores dedicados (*beowulf clusters*). Esse tipo de equipamento, entretanto, é caro e exige operação especializada. Por isso problemas paralelos de logística somente ambiente científico ou em departamentos de pesquisa e desenvolvimento de grandes

Por outro lado, as organizações usualmente dispõem de centenas ou mesmo milhares de computadores, interligados por redes locais de boa qualidade, e que passam grande parte de seu tempo ocioso, por exemplo, à noite e aos fins-de-semana.

A utilização coordenada desses recursos computacionais permitiria construir um supercomputador virtual, com capacidade de processamento paralelo suficiente para solucionar problemas complexos de interesse para essas organizações, como é o caso de problemas de VMI de grande escala.

Infra-estruturas computacionais que utilizam recursos heterogêneos distribuídos são cada vez mais utilizadas como plataforma para aplicação particular, destacam-se os projetos *Tonic Scientific Computing* [7] e *Cluster Adaptativo*

[8]. Por utilizarem apenas componentes de software Java, e pela sua semelhança operacional com *beowulf clusters*, essas infra-estruturas também são conhecidas por *JavaWulf* [9].

A grande vantagem em se utilizar apenas componentes de tecnologia Java é a simplicidade na geração e manutenção do código distribuído. O trabalho de configuração local dos nós trabalhadores em redes heterogêneas é reduzido, necessário apenas instalar o ambiente de *runtime* de Java e adicionar um aplicativo trabalhador que é executado apenas quando o computador está ocioso.

Neste trabalho é utilizada a infra-estrutura computacional tipo *JavaWulf* descrita em [10] e que foi implementada no CIETEC – Centro Incubador de Empresas Tecnológicas que se localiza no IPEN – Instituto de Pesquisas Energéticas e Nucleares dentro do campus da USP – Universidade de São Paulo.

No CIETEC estão instaladas cerca de 100 empresas de base tecnológica que compartilham uma infra-estrutura de rede padrão Ethernet de 100 Mbps. Há uma ampla variedade de equipamentos interconectados: PCs e notebooks com versões diferentes de Windows (XP, 2000, 98, 95), PCs Linux com distribuições variadas (Debian, RPM) e estações de trabalho rodando MacOS. A estação servidora foi instalada num PC com CPU Intel 2.53MHz, 1 GB memória RAM.

A figura 4 mostra a disponibilidade do *JavaWulf* durante uma semana de operação, de segunda a sexta-feira. Cada uma das 5 curvas mostradas na figura 4 refere-se a um dia útil da semana (segunda-feira, terça-feira, quarta-feira, quinta-feira e sexta-feira).

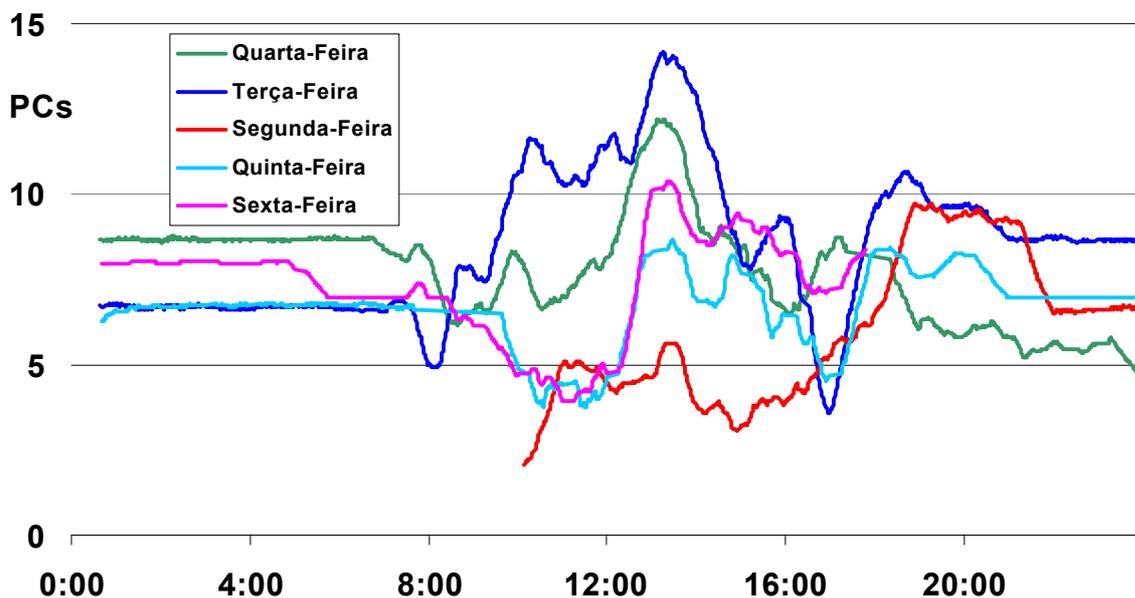


Figura 4. Disponibilidade de PCs no *JavaWulf* de segunda-feira a sexta-feira.

As curvas apresentam basicamente o mesmo padrão de comportamento em qualquer dia da semana. No período diurno (7:00-19:00) o número de PCs disponíveis varia bastante, ocorrendo um pico de disponibilidade na hora do almoço (12:00-14:00). Por outro lado, no período noturno (19:00-7:00) o número de PCs disponíveis é menor porém constante. Esse comportamento ocorre porque cada equipamento só se conecta ao *JavaWulf* se estiver ocioso por pelo menos 1 minuto. No período diurno há um

PCs são desligados, mas os demais operam no JavaWulf sem interrupções.

5. Experimentos

Os experimentos com o modelo de VMI desenvolvido nas seções 2 e 3 foram realizados com a infra-estrutura computacional descrita na seção 4 acima.

O exemplo de aplicação gerado foi baseado na cidade de São Paulo. As coordenadas geográficas (latitude e longitude) foram sorteadas aleatoriamente com distribuição uniforme dentro do perímetro da cidade. Foram utilizados adimensionais relacionados de forma coerente entre si, e representando situações reais de operação. Os valores de capacidade máxima de estoque, estoque inicial e taxa de consumo por unidade de tempo para cada local de reabastecimento foram sorteados utilizando distribuições uniformes. As distâncias foram estimadas pela distância euclidiana entre os locais vezes um fator para corrigir curvaturas. Admitiu-se velocidade média constante em todos os trajetos e em qualquer período do dia. A capacidade de cada veículo da frota foi admitida fixa e igual para todos os veículos (frota homogênea). Essas restrições foram impostas para simplificar a análise de modelo de grande escala.

A figura 5 mostra os resultados obtidos com o JavaWulf para o modelo VMI para planejamento de 10 entregas seqüenciais para 4.000 locais de reabastecimento. O modelo foi rodado em 3 situações: (a) um único PC (Pentium III 500MHz 256MB RAM), (b) JavaWulf no período diurno e (c) JavaWulf no período noturno.

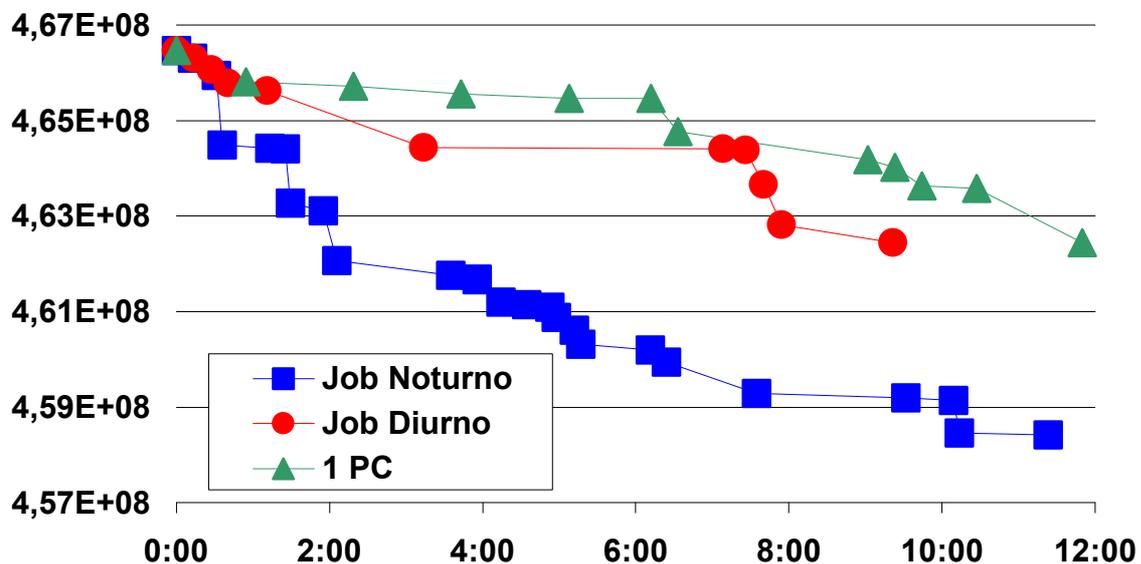


Figura 5. Variação da função objetivo com o tempo de processamento.

Os 3 experimentos mostrados na figura 5 partiram da mesma solução inicial (variabilidade = $466,46 \times 10^6$ e frota de 389 veículos). Após 12 horas de execução, o experimento rodado a noite forneceu a melhor solução (variabilidade = $458,42 \times 10^6$ e frota de 353 veículos). O *job* diurno apresentou resultados apenas marginalmente melhores que o experimento com apenas um único PC.

Confrontando-se os resultados da figura 5 com as curvas de disponibilidade dos PCs do JavaWulf apresentadas na figura 4, seria de se esperar que o processamento diurno fornecesse melhores resultados do que os apresentados pois há um grande número de PCs disponíveis. No entanto a disponibilidade de PCs no período diurno não é contínua devido à frequente entrada e saída de PCs no JavaWulf, em função do uso do equipamento pelo usuário que interrompe sua ociosidade.

No *job* noturno há um menor número de PCs disponíveis, mas que estão ininterruptamente disponíveis para o *job*. A operação intermitente (*stop-and-go*) no período diurno acaba por penalizar significativamente o processamento do experimento, tornando-o pouco melhor que o processamento em um único PC. Isso ocorre devido ao tamanho do modelo (grande escala), para o qual o tempo de configuração para início do *job* é relativamente grande.

6. Conclusões e Aplicações na Indústria Siderúrgica

O método proposto para modelamento de VMI utilizando técnicas metaheurísticas paralelas utilizando uma infra-estrutura computacional de alto desempenho, descritos nas seções anteriores, permite solucionar modelos VMI de grande escala em tempo total de processamento compatível com a operação diária de planejamento logístico.

Esse modelo é uma ferramenta que pode fornecer bons indicadores para o planejamento de rotas de abastecimento. O escopo de análise ampliou-se significativamente, de algumas dezenas de clientes para milhares. O horizonte de planejamento passa de dias ou semanas para um período virtualmente infinito.

O JavaWulf mostrou-se uma ferramenta de grande importância para a análise de modelos de grande escala. Foi possível realizar diversos testes com o modelo e analisar a influência de parâmetros do modelo na solução do problema. Os resultados obtidos com o JavaWulf são coerentes com os resultados obtidos anteriormente mas que utilizavam recursos computacionais sofisticados (e caros) como estações de trabalho multiprocessadas de alto desempenho [3] e um *beowulf cluster* [5]. A vantagem econômica de um Javawulf é evidente. A utilização dos recursos ociosos de redes locais de PCs e a simplicidade operacional do JavaWulf viabiliza o uso de computação paralela na solução de problemas logísticos do dia-a-dia das empresas.

Essa metodologia pode ser aplicada na indústria siderúrgica trazendo benefícios em praticamente todas as etapas da sua cadeia de suprimentos:

Clientes finais. O monitoramento dos estoques numa grande base de clientes poderá melhorar o atendimento aos clientes finais, detectando-se instantaneamente falta de produtos. Isso é particularmente interessante no fornecimento para indústria automobilística que atua com sistemas de entrega *just-in-time*. Novos mercados que exigem um controle capilar podem ser explorados (p.ex. construção civil).

Distribuição. Com uma melhor programação de entregas, a frota pode ser minimizada e menores custos finais aos clientes podem ser obtidos, melhorando a margem de lucro. Pode-se redesenhar a malha de distribuição, reduzindo-se custos de instalações e operações, sem prejudicar o abastecimento no usuário final.

Planejamento da produção. O planejamento da produção fica mais ágil e voltado às necessidades dos clientes. Não se produz itens que não tenham demanda. Há menos estoque de produtos semi-acabados (*work-in-progress*).

7. Agradecimentos

Agradecemos pelas discussões técnicas sobre metaheurísticas com as Profas. Dras. Cíntia Rigão Scrich (UNIP/Campinas) e Lúcia F. de Almeida Guimarães (PUC/Campinas), sobre processamento paralelo e distribuído com os Profs. Drs. Alfredo Goldman (IME/USP) e Fábio Kon (IME/USP) e sobre modelamento logístico com o Prof. Dr. Marco Antônio Brinati (POLI/USP).

Agradecemos também ao CIETEC (www.cietec.org.br) e às empresas incubadas CNZ Engenharia e Informática (www.cnz.com.br), Livronline e-Learning (www.livronline.com.br), Mamute Mídia (www.mamutemidia.com.br) e Sala Viva (www.salaviva.com.br) que gentilmente cederam seus equipamentos para utilização experimental na infraestrutura computacional.

8. Referências Bibliográficas

- [1] CAMPBELL, A., CLARKE, L., SAVELSBERGH, M., *An inventory routing problem*. The Logistics Institute, Georgia Institute of Technology, Abril, 1999.
- [2] KLEYWEGT, A. NORI, V., SAVELSBERGH, M. *The stochastic inventory routing problem with Direct Deliveries*. School of Industrial and Systems Engineering, Georgia Institute of Technology, Junho, 1999.
- [3] STANDERSKI, N. *Aplicação de técnicas de computação paralela na solução de problemas de otimização em logística*. Relatório Técnico Projeto 01/03196-6 FAPESP, São Paulo, Fevereiro, 2002.
- [4] CUNG, V.-D., MARTINS, S.L., RIBEIRO, C.C., ROUCARIOL, C. *Strategies for the parallel implementation of metaheuristics*. in Essays and Surveys in Metaheuristics (C.C. Ribeiro e P. Hansen editores), 263-308, Kluwer, 2001.
- [5] STANDERSKI, N. *Aplicação de algoritmos genéticos paralelos a problemas de grande escala de VMI – Vendor Managed Inventory*, XXXV Simpósio Brasileiro de Pesquisa Operacional, Natal, Novembro, 2003.
- [6] VAN KEMENADE, C.H.M., KOK, J.N. *An evolutionary approach to time constrained routing problems*. CWI, Centrum voor Wiskunde Informatica, Amsterdã, Holanda, 1995, ISSN 0169-118X, 12p.
- [7] NOBLE, M. *Tonic, A Java TupleSpaces Benchmark Project*, 2003. (<http://he-www.harvard.edu/~mnoble/tonic/doc/>).
- [8] BATHEJA, J. e PARASHAR, M. *A Framework for Adaptive Cluster Computing using JavaSpaces*, The Applied Software Systems Laboratory, Department of Electrical and Computer Engineering, The State University of New Jersey. (<http://www.caip.rutgers.edu/TASSL/Papers/jinihpc-hpcn01.pdf>).
- [9] MARCHAND, B. *JavaWulf*, Fifth Jini Technology Community Meeting, Amsterdam, 2000. (<http://www.jini.org/meetings/fifth/slides/JavaWulf/JavaWulf.htm>).
- [10] STANDERSKI, N. *Construção de um Cluster JavaWulf para Aplicação a Problemas Logísticos de VMI de Grande Escala*. Relatório Técnico Projeto 01/03196-6 FAPESP, São Paulo, Maio, 2004.

A METHODOLOGY TO SOLVE LARGE-SCALE VMI - VENDOR MANAGED INVENTORY PROBLEMS

Dr. Nelson Standerski

Abstract

This paper proposes a methodology based on parallel meta-heuristics and high performance computing to find solutions of large-scale vendor managed inventory (VMI) problems. The model assumes direct deliveries to a large customer basis. Each customer is defined by its geographical coordinates, maximum storing capacity, uniform consume rate, and initial inventory level. The model was developed using the classical linear deterministic formulation of optimal inventory re-supply and parallel genetic algorithms.

The solution of the model is performed with the help of a distributed computational framework implemented on a local area PC network. The framework is based on java components, and operates only during the idle time of the PCs. The delivered computation power of the framework is similar to a dedicated PC cluster.

Experiments made with the VMI model and the computational framework showed that it is possible to run models with up to 4.000 re-supply points. The model calculation time decreases proportional to the increase of the number of available PCs.

In this way, the proposed methodology broads the analysis of VMI problems from dozens of customers to thousands. The computational framework is of great economic advantage, as it only makes use of idle computational resources.

Keywords: *logistics, VMI, parallel genetic algorithms.*

XXIII International Logistics Seminar - Associação Brasileira de Metalurgia e Materiais – Belo Horizonte, June, 16 to 18, 2004.

Dr. Nelson Standerski received his degrees of Engineer and MSc from the University of São Paulo (Brazil), and PhD from the Technical University of Berlin (Germany). He worked at IPT - Technological Research Institute of São Paulo, and at Siemens Brazil. Since 1995 he works as senior consultant of Paperless P&D Ltda (www.paperless.com.br). E-mail: nelson@paperless.com.br

This work has the support of *FAPESP – Fundação de Amparo a Pesquisa do Estado de São Paulo* (contract # 01/03196-6).