

PROBLEMA INTEGRADO DE PRODUÇÃO E ROTEAMENTO COM FROTA HETEROGÊNEA: UMA ANÁLISE DE DUAS ABORDAGENS DE OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

Yelky Castro Jimenez

Universidade Estadual de Campinas
R. Pedro Zaccaria, 1300, Limeira - SP, 13484-350
y237753@dac.unicamp.br

Diego Jacinto Fiorotto

Universidade Estadual de Campinas
R. Pedro Zaccaria, 1300, Limeira - SP, 13484-350
fiorotto@unicamp.br

RESUMO

Abordamos o problema integrado de roteamento e produção em um ambiente de uma única planta de produção, um único produto, sob a política de estoque nível máximo (*Maximum Level*) e com uma frota heterogênea em um horizonte de planejamento finito. Propomos duas abordagens de solução: a primeira, um algoritmo de otimização por enxame de partículas; e a segunda, uma abordagem híbrida que combina a otimização por enxame de partículas y uma heurística construtiva para a geração de soluções no problema de roteamento de veículos. Testamos essas metodologias em instâncias da literatura, constatando que a abordagem híbrida oferece soluções de melhor qualidade.

PALAVRAS CHAVE. Problema integrado de produção e roteamento, otimização por exame de partículas, Vizinho mas próximo.

OC – Otimização Combinatória, PM - Programação Matemática

ABSTRACT

We address the production routing problem in a single-plant, single-product environment, under the Maximum Level inventory policy, and with a heterogeneous fleet over a finite planning horizon. We propose two solution approaches: the first one is a particle swarm optimization algorithm; and the second is a hybrid approach that combines particle swarm optimization with a constructive heuristic for generating solutions to the vehicle routing problem. We tested these methodologies on instances from the literature, finding that the hybrid approach provides higher quality solutions.

KEYWORDS. Production Routing Problem. Particle Swarm Optimization. Nearest Neighbor.

CO - Combinatorial Optimization, MP - Mathematical Programming

1. Introdução

As decisões de produção, inventário e distribuição são essenciais na cadeia de suprimentos. Otimizar esses processos reduz custos, aumenta a eficiência e melhora a satisfação do cliente. Empresas como Kellogg Brown et al. [2001] e Frito-Lay Çetinkaya et al. [2009] obtiveram economias milionárias com uma otimização integrada. No entanto, a complexidade desses problemas integrados, classificados como NP-hard, requer soluções eficientes. Algoritmos avançados, como a otimização por enxame de partículas, são promissores. A integração das decisões de produção, inventário, distribuição e roteamento pode economizar entre 3% e 20% nos custos, segundo estudos de Chandra e Fisher [1994]. Nesta pesquisa, abordamos o problema integrado de roteamento e produção utilizando duas abordagens: otimização por enxame de partículas e uma híbrida que combina este algoritmo e uma heurística construtiva para às decisões de roteamento. Testamos nossas metodologias em instâncias propostas por Adulyasak et al. [2014], constatando que a abordagem híbrida oferece soluções de melhor qualidade em menor tempo.

2. Notação matemática e modelagem proposta

A modelagem apresentada é uma adaptação baseada na proposta de Bard e Nananukul [2010]). Ela é formulada com variáveis contínuas para representar as decisões de dimensionamento de lotes e gestão de inventário, variáveis inteiras para controlar as quantidades entregues por meio de uma frota heterogênea e variáveis binárias para o processo de planejamento de rotas. Em um grafo dirigido e completo denotado por $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, onde $\mathcal{N} = \{0, 1, \dots, |\mathcal{N}|\}$ é o conjunto de nós da rede e $\mathcal{A} = \{(i, j) \mid \forall i \in \mathcal{N}, j \in \mathcal{N} \mid i \neq j\}$ é o conjunto de todos os possíveis arcos que compõem a rede. A fábrica, representada pelo nó 0, deve fornecer e satisfazer a demanda dos clientes $i \in \mathcal{N}_c$, denotados pelo conjunto $\mathcal{N}_c = \{1, 2, \dots, |\mathcal{N}|\}$, em um horizonte de planejamento finito $\mathcal{T} = \{1, 2, \dots, |\mathcal{T}|\}$, onde as entregas serão realizadas por uma frota de veículos $\mathcal{K} = \{1, 2, \dots, |\mathcal{K}|\}$ com capacidades diferentes. O objetivo é minimizar os custos de produção, estoque, distribuição e roteamento.

Notação	Definição
u	Custo unitário de produção
f	Custo fixo de configuração da produção
c	Capacidade da produção
Q_k	Capacidade do veículo $k \in \mathcal{K}$
h_i	Custo de estocar uma unidade no nó $i \in \mathcal{N}$
l_i	Nível de estoque máximo no nó $i \in \mathcal{N}$
I_{0i}	Estoque inicial no nó $i \in \mathcal{N}$
ct_{ij}	Custo de percorrer a aresta $(i, j) \in \mathcal{A}$
d_{it}	Demanda do cliente $i \in \mathcal{N}_c$ no período $t \in \mathcal{T}$
M_t	$\min\{c, \sum_{j=t}^{ \mathcal{T} } \sum_{i \in \mathcal{N}_c} d_{i,j}\} \forall t \in \mathcal{T}$
\tilde{M}_{it}	$\min\{l_i, \min\{Q_k\} \forall k \in \mathcal{K}, \sum_{j=t}^{ \mathcal{T} } d_{i,j}\} \forall t \in \mathcal{T}, i \in \mathcal{N}_c$
p_t	Quantidade produzida no período $t \in \mathcal{T}$
y_t	1, se tem produção na fábrica no período $t \in \mathcal{T}$. 0, caso contrário.
b_{kt}	1, se o veículo $k \in \mathcal{K}$ for utilizado no período $t \in \mathcal{T}$. 0, caso contrário.
I_{it}	Quantidade de unidades a estocar no nó $i \in \mathcal{N}$ no período $t \in \mathcal{T}$
q_{it}	Quantidades de unidades a serem enviados no cliente $i \in \mathcal{N}_c$ no período $t \in \mathcal{T}$
s_{it}	1, se o cliente $i \in \mathcal{N}_c$ for visitado no período $t \in \mathcal{T}$. 0, caso contrário.
z_{ikt}	1, se o cliente $i \in \mathcal{N}_c$ for visitado pelo veículo $k \in \mathcal{K}$ no período $t \in \mathcal{T}$. 0, caso contrário.
w_{ikt}	Carga de um veículo $k \in \mathcal{K}$ antes de fazer entrega ao cliente $i \in \mathcal{N}_c$ no período $t \in \mathcal{T}$
x_{ijkt}	1, se o veículo $k \in \mathcal{K}$ percorre o arco $(i, j) \in \mathcal{A}$ no período $t \in \mathcal{T}$. 0, caso contrário.

Tabela 1: Notação matemática

Função Objetivo: Minimizar o custo total

$$\min \left(\sum_{t \in \mathcal{T}} (u \cdot p_t) + \sum_{t \in \mathcal{T}} (f \cdot y_t) + \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}} (h_i \cdot I_{it}) + \sum_{t \in \mathcal{T}} \sum_{(i,j) \in \mathcal{A}} \sum_{k \in \mathcal{K}} (ct_{ij} \cdot x_{ijkt}) \right) \quad (1)$$

Restrições:

$$I_{0(t-1)} + p_t = I_{0t} + \sum_{i \in \mathcal{N}_c} q_{it} \quad \forall t \in \mathcal{T} \quad (2)$$

$$I_{i(t-1)} + q_{it} = d_{it} + I_{it} \quad \forall t \in \mathcal{T}, i \in \mathcal{N}_c \quad (3)$$

$$p_t \leq M_t \cdot y_t \quad \forall t \in \mathcal{T} \quad (4)$$

$$I_{0t} + \sum_{i \in \mathcal{N}_c} q_{it} \leq l_0 \quad \forall t \in \mathcal{T} \quad (5)$$

$$I_{i(t-1)} + q_{it} - d_{it} \leq l_i \quad \forall t \in \mathcal{T}, i \in \mathcal{N}_c \quad (6)$$

$$s_{it} \leq q_{it} \leq \tilde{M}_{it} \cdot s_{it} \quad \forall t \in \mathcal{T}, i \in \mathcal{N}_c \quad (7)$$

$$z_{ikt} \leq s_{it} \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N}_c \quad (8)$$

$$z_{ikt} \leq b_{kt} \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N}_c \quad (9)$$

$$\sum_{j \in \mathcal{N} \mid j \neq 0} x_{0jkt} = b_{kt} \quad \forall t \in \mathcal{T}, k \in \mathcal{K} \quad (10)$$

$$\sum_{i \in \mathcal{N} | i \neq 0} x_{i0kt} = b_{kt} \quad \forall t \in \mathcal{T}, k \in \mathcal{K} \quad (11)$$

$$\sum_{j \in \mathcal{N} | i \neq j} \sum_{k \in \mathcal{K}} x_{ijk t} = s_{it} \quad \forall t \in \mathcal{T}, i \in \mathcal{N}_c \quad (12)$$

$$\sum_{i \in \mathcal{N} | i \neq j} \sum_{k \in \mathcal{K}} x_{ijk t} = s_{jt} \quad \forall t \in \mathcal{T}, j \in \mathcal{N}_c \quad (13)$$

$$\sum_{j \in \mathcal{N} | i \neq j} x_{jikt} + \sum_{j \in \mathcal{N} | i \neq j} x_{ijk t} = 2 \cdot z_{ikt} \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N}_c \quad (14)$$

$$w_{ikt} - w_{jkt} \geq q_{it} - \tilde{M}_{it}(1 - x_{ijk t}) \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, (i, j) \in \mathcal{A} | i \neq 0 \wedge j \neq 0 \quad (15)$$

$$w_{ikt} \leq Q_k \cdot z_{ikt} \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N}_c \quad (16)$$

$$\sum_{k \in \mathcal{K}} z_{ikt} \leq s_{it} \quad \forall t \in \mathcal{T}, i \in \mathcal{N}_c \quad (17)$$

$$\begin{aligned}
 p_t &\geq 0, y_t \in \{0, 1\} \quad \forall t \in \mathcal{T} \\
 b_{kt} &\in \{0, 1\} \quad \forall t \in \mathcal{T}, k \in \mathcal{K} \\
 I_{it} &\geq 0, q_{it} \geq 0, s_{it} \in \{0, 1\} \quad \forall t \in \mathcal{T}, i \in \mathcal{N}_c \\
 w_{ikt} &\geq 0, z_{ikt} \in \{0, 1\} \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N}_c \\
 x_{ijk t} &\in \{0, 1\} \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, (i, j) \in \mathcal{A}
 \end{aligned} \quad (18)$$

A expressão 1 define a função objetivo do problema, minimizando os custos de produção, configuração da produção, inventário e roteirização. As restrições estão agrupadas da seguinte forma: da 2 à 6, restrições relacionadas ao problema de produção; da 7 à 9, restrições de acoplamento entre os dois problemas; da 10 à 17, restrições associadas ao problema de roteirização de veículos e distribuição; e a 18 estabelece o domínio das variáveis.

3. Algoritmo proposto

O PSO pelas siglas em inglês (*Particle Swarm Optimization*) é um algoritmo de otimização baseado no comportamento de enxame. Foi proposto por Kennedy e Eberhart [1995] e se inspira no movimento coletivo de organismos como bandos de pássaros ou cardumes de peixes. O algoritmo foi codificado conforme às restrições mostradas no modelo da seção 2. Uma partícula é uma solução factível e completa do problema que armazena as quantidades produzidas (p_t) se houver produção nesse período (y_t), os estoques (I_{it}), a distribuição das entregas (q_{it}, s_{it}, z_{ikt}) e as informações relacionadas ao planejamento de rotas ($x_{ijk t}, w_{ikt}, b_{kt}$). Cada variável mantém sua natureza binária ou contínua conforme mostrado na tabela 1.

Cada partícula possui dois espaços reservados: o primeiro guarda sua melhor posição histórica, e o segundo guarda a posição atual. Dessa forma, a população é formada por um número de partículas dado como parâmetro. A população é inicializada colocando novas partículas tanto nos espaços reservados para as posições atuais quanto para as melhores posições históricas de cada partícula. Cada partícula é avaliada na função *fitness*, tanto na sua melhor posição histórica quanto na posição atual; a melhor avaliação será mantida como a melhor posição histórica na próxima iteração. Adicionalmente, em cada iteração, é guardada a informação da melhor partícula global.

As novas posições de cada partícula são orientadas pela velocidade baseada na inércia (posição atual) e nas componentes cognitivas (melhor histórico) e sociais (melhor global), conforme descrito por Kennedy e Eberhart [1995]. No processo de otimização e convergência do algoritmo para a solução ótima, algumas partículas podem perder factibilidade, devido à natureza do movimento das partículas. Quando isso ocorre, a melhor posição histórica, que a priori é uma solução factível, é mantida e uma nova partícula é integrada na nova posição. O critério de parada do algoritmo é dado pelo número de iterações.

Híbrido PSO-NNH

Notamos que, na construção da partícula, há um bom comportamento nas decisões de produção, inventário e distribuição, mas não no planejamento de rotas. Este híbrido conserva a geração de soluções para as decisões de produção, inventário e distribuição utilizando a geração inicial de partículas e constrói as rotas por meio de uma heurística construtiva NNH (*Nearest Neighbor Heuristic*) para determinar soluções no problema de

roteamento de veículos com múltiplos períodos. A heurística busca minimizar a distância total percorrida por cada veículo, assegurando que as entregas aos clientes (q_{it} sejam cumpridas dentro das capacidades dos veículos.

4. Experimentos computacionais e resultados

Para a implementação dos algoritmos, foi utilizada uma máquina equipada com um processador Core i7 de 13ª geração e 16 GB de RAM. O desenvolvimento ocorreu no ambiente de programação Python 3.8, utilizando bibliotecas específicas para otimização e processamento de dados. Além disso, foi empregado o software Gurobi Optimizer 10.0.3 para resolver problemas de otimização. As instâncias de teste foram retiradas da literatura da pesquisa de Adulyasak et al. [2014], todas com um horizonte de planejamento de 6 períodos e 3 veículos, variando o número de nós entre 15, 25 e 50, incluindo o nó associado à fábrica. Os experimentos realizados com Gurobi foram utilizados para encontrar limites inferiores e avaliar a qualidade das soluções propostas. Cada experimento em Gurobi teve um limite de tempo de 3600 segundos. Os resultados obtidos são apresentados na tabela 2.

Instancia	Gurobi			PSO			H-PSO-NNH			% Variação GAP
	Valor Fo	LI	GAP	Valor FO	GAP	Tempo	Valor FO	GAP	Tempo	
N15_I1	16725	16478	1,48%	20374	19,12%	44,05	18417	10,53%	11,05	-81,63%
N15_I2	15487	15487	0,00%	18632	16,88%	38,57	17126	9,57%	45,27	-76,38%
N15_I3	14438	13709	5,05%	16912	18,94%	6,99	15818	13,33%	15,75	-42,05%
N15_I4	17597	17526	0,40%	20551	14,72%	145,83	19331	9,34%	72,71	-57,64%
N15_I5	17434	16478	5,48%	20974	21,44%	14,16	18145	9,19%	61,32	-133,33%
N25_I1	24685	23511	4,76%	31582	25,56%	11,85	28090	16,30%	98,46	-56,77%
N25_I2	25591	24470	4,38%	32735	25,25%	28,44	28891	15,30%	239,46	-65,00%
N25_I3	52043	44616	14,27%	79231	43,69%	62,68	60120	25,79%	13,46	-69,41%
N25_I4	6857	5264	23,23%	13921	62,19%	83,91	9667	45,55%	35,46	-36,53%
N50_I1	46460	43925	5,46%	69371	36,68%	384,8	54889	19,97%	102,42	-83,64%
N50_I2	45992	40790	11,31%	66446	38,61%	19,3	51491	20,78%	484,71	-85,79%
N50_I3	40347	35849	11,15%	60680	40,92%	289,09	46251	22,49%	42,52	-81,95%
N50_I4	50453	39935	20,85%	60680	34,19%	139,41	51908	23,07%	283,58	-48,22%
N50_I5	41353	36716	11,21%	61717	40,51%	20,19	46812	21,57%	31,75	-90,83%

Tabela 2: Experimentos computacionais

Os resultados mostram que o PSO-NNH superou o PSO em termos de qualidade de solução e tempo de execução em todas as instâncias testadas. Especificamente, a abordagem híbrida PSO-NNH forneceu soluções com custos totais mais próximos aos obtidos pelo Gurobi e em um tempo significativamente menor. A solução do PSO, embora boa, foi menos eficiente na geração de rotas, o que impactou os custos totais.

5. Conclusões

O estudo abordou o problema integrado de produção e roteamento com uma frota heterogênea utilizando duas abordagens: o PSO e uma abordagem híbrida PSO-NNH. Os resultados dos experimentos mostraram que a abordagem híbrida proporciona soluções de melhor qualidade em um tempo menor, devido à combinação eficaz da otimização por enxame de partículas com a heurística construtiva de vizinho mais próximo para o problema de roteirização. A abordagem híbrida PSO-NNH demonstrou ser uma metodologia promissora para problemas complexos de produção e roteamento, oferecendo uma alternativa viável e eficiente para a solução desses problemas em ambientes de cadeias de suprimentos. Para trabalhos futuros, exploraremos outras heurísticas combinadas com PSO, bem como a aplicação desta abordagem em outros contextos de produção e distribuição.

Referências

- Adulyasak, Y., Cordeau, J. F., e Jans, R. (2014). Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science*, 48:20–45. ISSN 15265447.
- Bard, J. F. e Nananukul, N. (2010). A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers and Operations Research*, 37:2202–2217. ISSN 03050548.
- Brown, G., Keegan, J., Vigus, B., e Wood, K. (2001). The kellogg company optimizes production, inventory, and distribution. *Interfaces*, 31:1–15. ISSN 00922102.

- Chandra, P. e Fisher, M. L. (1994). Theory and methodology coordination of production and distribution planning *.
- Kennedy, J. e Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, p. 1942–1948 vol.4.
- Çetinkaya, S., Üster, H., Easwaran, G., e Keskin, B. B. (2009). An integrated outbound logistics model for frito-lay: Coordinating aggregate-level production and distribution decisions. *Interfaces*, 39:460–475.