Busca Local Iterada Aplicada à Minimização do Uso de Estoque Intermediário em Sistemas Industriais

Douglas Matuzalem Pontes Belo Lança
Marco Antonio M. Carvalho

Departamento de Computação
Instituto de Ciências Exatas e Biológicas
Universidade Federal de Ouro Preto

30 de agosto de 2017

UFOP
Universidade Federal de Ouro Preto

decom
departamento de computação

XLIX Simpósio Brasileiro de Pesquisa Operacional
Blumenau, SC - 27 a 30 de Agosto

Conteúdo

Contexto Industrial

- Uma única máquina produz diferentes produtos com demandas pré-definidas;
- A produção é dividida em estágios – a cada estágio um tipo de produto é fabricado;
- Diferentes clientes depositam ordens de compra;
- Estoque intermediário é utilizado para armazenar pedidos de compra incompletos;
- Pedidos de compra completamente produzidos são despachados.

Introdução

O **Espalhamento de Ordem** é o número de estágios entre a produção do primeiro produto de uma ordem de compra e a produção do último produto da mesma ordem.

O **Problema de Minimização do Espalhamento de Ordem** (*Minimization of Order Spread Problem*, MORP) visa minimizar o espalhamento máximo (ou médio) das ordens de compra por meio do sequenciamento da produção.

Motivação

- Trata-se de um problema NP-Difícil (Garey & Johnson, 1979);
- Aplicação prática industrial ampla;
- Literatura restrita;
- Equivalência com outros problemas da literatura, como o problema de minimização do comprimento de redes em circuitos eletrônicos (De Giovanni et al., 2013).

Introdução

Exemplo

	Painel	Batente	Rodapé	Sarrafo
Alice	✓	✗	✓	✗
Bob	✓	✓	✗	✓
Chico	✓	✓	✓	✗
Davi	✗	✗	✗	✓

Instância

- Conjunto J de produtos;
- Conjunto I de ordens de compra;
- Matriz P $(|I| \times |J|)$.

	j_1	j_2	j_3	j_4
i_1	1	1	0	0
i_2	1	0	1	0
i_3	0	0	0	1
i_4	0	0	0	1

Solução

Uma solução para o MORP é uma permutação π das colunas da matriz P, gerando uma matriz permutação Q^{π}.

Para identificar o comprimento dos espalhamentos de ordens a matriz Q^{π} é definida com a propriedade de 1*s consecutivos* de acordo com a equação abaixo:

$$q_{ij}^{\pi} = \begin{cases} 1 & \text{se } \exists x, \exists y \mid \pi[x] \leq j \leq \pi[y] \text{ e } p_{ix} = p_{iy} = 1 \\ 0, & \text{caso contrário} \end{cases} \tag{1}$$

	j_1	j_6	j_5	j_4	j_3	j_2
i_1	1	0	0	0	0	1
i_2	1	0	0	0	1	0
i_3	0	0	1	1	0	0
i_4	0	1	0	1	0	0
i_5	0	0	1	0	0	1
i_6	0	1	0	0	1	0

(a)

	j_1	j_6	j_5	j_4	j_3	j_2
i_1	1	1	1	1	1	1
i_2	1	1	1	1	1	0
i_3	0	0	1	1	0	0
i_4	0	1	1	1	0	0
i_5	0	0	1	1	1	1
i_6	0	1	1	1	1	0

(b)

Métodos Propostos

Visão Geral

A proposta deste trabalho é a implementação de um método em duas fases para solução do MORP:

- Na primeira, gera-se uma solução inicial por meio de uma heurística gulosa;
- Na segunda, aprimora-se a solução pela aplicação da metaheurística Busca Local Iterada (*Iterated Local Search*, ILS).

Solução Inicial

A solução inicial é gerada por uma heurística rápida:

- Pré-processamento: remoção de dados redundantes;
- Construção gulosa: método *Best Insertion*;

Solução Inicial: *Best Insertion*

Princípio

Partir de uma solução π vazia e adicionar cada um dos produtos, iterativamente, na melhor posição possível, até que se obtenha uma solução π completa.

Exemplo

	j_2	j_1
i_1	1	1
i_2	0	1
i_3	0	0
i_4	0	0
i_5	1	1
i_6	0	0

(a)

	j_1	j_2
i_1	1	1
i_2	1	0
i_3	0	0
i_4	0	0
i_5	1	1
i_6	0	0

(b)

Solução Inicial: *Best Insertion*

Exemplo

	j_3	j_2	j_1
i_1	0	1	1
i_2	1	0	1
i_3	0	0	0
i_4	0	0	0
i_5	0	1	1
i_6	1	0	0

(a)

	j_2	j_3	j_1
i_1	1	0	1
i_2	0	1	1
i_3	0	0	0
i_4	0	0	0
i_5	1	0	1
i_6	0	1	0

(b)

	j_2	j_1	j_3
i_1	1	1	0
i_2	0	1	1
i_3	0	0	0
i_4	0	0	0
i_5	1	1	0
i_6	0	0	1

(c)

Busca Local Iterada

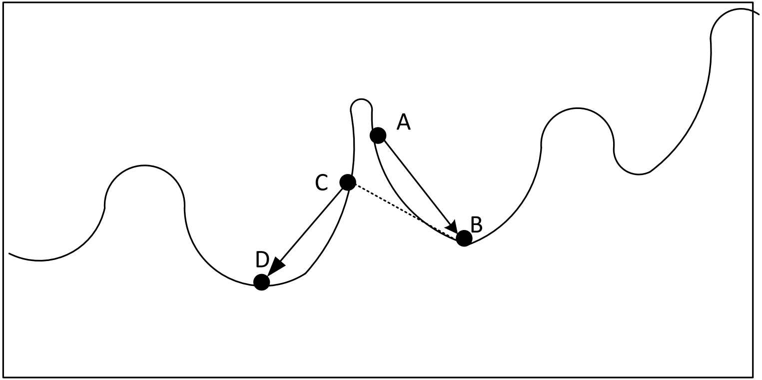
Descrição

- Proposta por Lourenço et al. (2003);
- Parte de uma solução inicial viável;
- Alterna iterativamente etapas de intensificação e diversificação da busca no espaço de soluções;
- Tende a não ficar presa em ótimos locais;

ILS Aplicada ao MORP

- Solução inicial heurística;
- Buscas locais (100%): 2-*swap* e agrupamento de 1-*blocks* (Paiva e Carvalho, 2017);
- Perturbação (20%): 2-opt;
- Critério de parada: 100 iterações.

Busca Local Iterada



Busca Local: 2-*swap*

Princípio

Trocar sucessivamente dois elementos de posição, mantendo as trocas que aprimorem a solução. A cada melhoria, o método é reiniciado.

Exemplo

	j_2	j_3	j_1	j_4	j_5	j_6
i_1	1	0	1	0	1	0
i_2	0	1	1	0	1	0
i_3	1	0	0	1	1	0
i_4	0	0	0	1	0	1
i_5	1	0	1	1	0	1
i_6	0	1	0	0	0	1

(a)

	j_5	j_3	j_1	j_4	j_2	j_6
i_1	1	0	1	0	1	0
i_2	1	1	1	0	0	0
i_3	1	0	0	1	1	0
i_4	0	0	0	1	0	1
i_5	0	0	1	1	1	1
i_6	0	1	0	0	0	1

(b)

Busca Local: Agrupamento de *1-blocks*

Princípio

Movimentar as colunas de uma matriz binária de maneira a agrupar os elementos não nulos em uma linha específica.

Exemplo

	j_1	j_2	j_3	j_4	j_5	j_6
i_1	1	1	0	0	1	0
i_2	1	0	1	0	1	0
i_3	0	1	0	1	1	0
i_4	0	0	0	1	0	1
i_5	**1**	**1**	0	**1**	**1**	0
i_6	0	0	1	0	0	1

Busca Local: Agrupamento de *1-blocks*

Princípio

Movimentar as colunas de uma matriz binária de maneira a agrupar os elementos não nulos em uma linha específica.

Exemplo

	j_1	j_3	j_2	j_4	j_5	j_6
i_1	1	0	1	0	1	0
i_2	1	1	0	0	1	0
i_3	0	0	1	1	1	0
i_4	0	0	0	1	0	1
i_5	**1**	0	**1**	**1**	**1**	0
i_6	0	1	0	0	0	1

(a)

	j_1	j_3	j_4	j_5	j_2	j_6
i_1	1	0	0	1	1	0
i_2	1	1	0	1	0	0
i_3	0	0	1	1	1	0
i_4	0	0	1	0	0	1
i_5	**1**	0	**1**	**1**	**1**	0
i_6	0	1	0	0	0	1

(b)

Busca Local: Agrupamento de *1-blocks*

Exemplo

	j_3	j_1	j_4	j_5	j_2	j_6
i_1	0	1	0	0	1	0
i_2	1	1	0	1	0	0
i_3	0	0	1	1	1	0
i_4	0	0	1	0	0	1
i_5	0	**1**	**1**	**1**	**1**	0
i_6	1	0	0	0	0	1

(a)

	j_3	j_4	j_5	j_2	j_1	j_6
i_1	0	0	0	1	1	0
i_2	1	0	1	0	1	0
i_3	0	1	1	1	0	0
i_4	0	1	0	0	0	1
i_5	0	**1**	**1**	**1**	**1**	0
i_6	1	0	0	0	0	1

(b)

Perturbação: 2-opt

Princípio

Inverter os elementos dentro de um intervalo definido aleatoriamente.

Exemplo

	j_2	j_3	j_1	j_4	j_5	j_6
i_1	1	0	1	0	1	0
i_2	0	1	1	0	1	0
i_3	1	0	0	1	1	0
i_4	0	0	0	1	0	1
i_5	1	0	1	1	0	1
i_6	0	1	0	0	0	1

(a)

	j_4	j_1	j_3	j_2	j_5	j_6
i_1	0	1	0	1	1	0
i_2	0	1	1	0	1	0
i_3	1	0	0	1	1	0
i_4	1	0	0	0	0	1
i_5	1	1	0	1	0	1
i_6	0	0	1	0	0	1

(b)

Experimentos Computacionais 1

Instâncias MORP de Fink & Voss (1999)

Conjunto de 797 instâncias artificiais para o MORP é dividido em 8 grupos, com aproximadamente 100 instâncias cada, geradas aleatoriamente.

Comparação

Estado da arte relacionado ao MORP: Busca Tabu Reativa (TSRE5000) de Fink & Voss (1999).

Setup

- Métodos implementados utilizando a linguagem C++ e compilados utilizando g++ 4.4.1 e a opção de otimização -O3;
- Computador Intel i5 Quad Core de 3.2GHz, 8 GB de RAM, e sistema operacional Ubuntu 15.10.

m	v	*TSRE5000* BKS	*TSRE5000* T (s)	*ILS-MORP* S	*ILS-MORP* S^*	*ILS-MORP* gap(%)	*ILS-MORP* σ	*ILS-MORP* T (s)
50	0,25	14,04	142,6	14,90	14,31	1,92	0,49	0,48
50	0,50	12,43	79,5	15,03	14,56	17,13	0,30	0,44
50	0,75	4,87	58,6	3,04	2,71	-37,57	0,24	0,41
50	1,00	1,38	35,1	1,48	1,29	-6,52	0,19	0,70
60	0,25	16,41	159,5	18,18	17,44	6,27	0,54	0,97
60	0,50	14,61	142,6	16,84	16,35	11,90	0,35	0,80
60	0,75	5,33	98,2	4,18	3,79	-28,89	0,31	0,92
60	1,00	1,45	58,9	1,66	1,32	-8,96	0,29	0,83

Experimentos Computacionais 2

Instâncias GMLP e MOSP

Diferentes conjuntos com um total de 68 instâncias reais e artificiais para o problema de minimização do comprimento de redes em circuitos eletrônicos e equivalentes.

Número de padrões $m \in \{20, 50, 75, 100, 141\}$.

Comparação

Algoritmo genético (GA) de De Giovanni et al. (2013).

Setup

Computador Intel i5 Quad Core de 3.2GHz, 8 GB de RAM, e sistema operacional Ubuntu 15.10.

Experimentos Computacionais

Resumo

- Soluções iguais em 10 instâncias;
- ILS obteve melhores soluções em 30 instâncias;
- GA obteve melhores soluções em 28 instâncias;
- *gap* médio de 3,18%;
- Melhoria de 34,67% da solução inicial em média;
- Tempo de execução máximo de 23,27 s;
- Convergência média em 45,02% do tempo de execução.

Conclusões

- O MORP é um importante problema no contexto industrial, que pode auxiliar no ritmo de atendimento das ordens de compra e melhor utilização de estoque intermediário;
- Existem poucas propostas na literatura para a resolução do MORP;
- O ILS proposto gerou novos melhores resultados para parte das instâncias, especificamente as esparsas;
- Trabalhos futuros se concentrarão em aprimorar o algoritmo proposto a fim de se obter melhores resultados para instâncias densas.

Agradecimentos

Apoio

- Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq);
- Fundação de Apoio à Pesquisa do Estado de Minas Gerais (FAPEMIG); e
- Universidade Federal de Ouro Preto (UFOP).

Perguntas?


thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
спасибо
Баярлалаа
рахмат
dank je
bedankt
obrigado
dziękuje
sukriya
kop khun krap
go raibh maith agat
mochchakkeram
tapadh leat
arigatō
dakujem
grazie
merci
terima kasih
감사합니다
ありがとう
ευχαριστώ
мерси
хвала
sagolun
hvala
mauruuru
faafetai lava
kiitos
dankie
dhanyavad
vinaka
blagodaram
mersi
barka
welalin
tack
misaotra
matondo
paldies
grazzi
mahalo
asante
manana
obrigada
tenki
djiere dieuf
mamnun
chokrane
trugarez
chnorakaloutioun
gratias ago
gracies
sulpay
sobodi
dekuji
mesi
najis tuke
kam sah hamnida
tanemirt
rahmet
diolch
dhanyavadagalu
shukriya
xiexie
enkosi
takk
taiku
akun
dankon
koszonom
nandri
gracie
bayarlalaa
didi madloba
rahmat
дякую