



De 11 a 14 - NOVEMBRO DE 2024

**Inteligência Artificial
na Gestão de Operações:**
Limitações e possibilidades



UM MÉTODO DE DECOMPOSIÇÃO PARA O PROBLEMA DE DESIGNAÇÃO DE OPERAÇÕES A MÁQUINAS

José Francisco Ferreira Ribeiro - jffr@fearp.usp.br
UNIVERSIDADE DE SÃO PAULO - USP – FEARP RIBEIRÃO PRETO

ÁREA: 1. ENGENHARIA DE OPERAÇÕES E PROCESSOS DA PRODUÇÃO

SUBÁREA: 1.2 – PLANEJAMENTO, PROGRAMAÇÃO E CONTROLE DA PRODUÇÃO

RESUMO: NESTE ARTIGO É APRESENTADO UM MÉTODO DE DECOMPOSIÇÃO EM DUAS ETAPAS PARA EFETUAR A DEFINIÇÃO DAS MÁQUINAS PARA FABRICAR AS PEÇAS, E A DESIGNAÇÃO DAS OPERAÇÕES ÀS MÁQUINAS ESCOLHIDAS. NA PRIMEIRA ETAPA, É DEFINIDO UM CONJUNTO DE SOLUÇÕES FACTÍVEIS DE UM PROBLEMA DA MOCHILA ATRAVÉS DE PROGRAMAÇÃO DINÂMICA OU DE UM PROCEDIMENTO HEURÍSTICO. NA SEGUNDA ETAPA, RESOLVE-SE UM PROBLEMA DA DESIGNAÇÃO USANDO O ALGORITMO DE ENUMERAÇÃO IMPLÍCITA OU O ALGORITMO GULOSO. O MÉTODO PROPOSTO E OS ALGORITMOS DESENVOLVIDOS FORAM IMPLEMENTADOS EM LINGUAGEM C. O PROGRAMA ESTÁ IMPLANTADO EM MICROCOMPUTADOR E PODE SER UTILIZADO EM TEMPO REAL.

PALAVRAS-CHAVES: PROBLEMA DA DESIGNAÇÃO, PROBLEMA DA MOCHILA, PROGRAMAÇÃO DINÂMICA, ENUMERAÇÃO IMPLÍCITA, HEURÍSTICAS.

1. INTRODUÇÃO

Neste artigo é apresentado um método de decomposição para definir as máquinas a executar as operações sobre os lotes de peças em um sistema de fabricação “job shop”, e para designar as operações a executar sobre as peças às máquinas da combinação definida na etapa precedente. O método de decomposição é composto de duas etapas:

1) Etapa de definição das máquinas, onde é resolvido um problema da mochila (CACCHIANI et al., 2022) por programação dinâmica (ou procedimento heurístico) para gerar e classificar combinações de máquinas capazes de cobrir a totalidade da carga de trabalho existente sobre os tipos de máquinas.

2) Etapa de designação das peças às máquinas, onde é resolvido um modelo matemático em variáveis bivalentes (MANIEZZO et al., 2021) por enumeração implícita (ou procedimento heurístico) para algumas combinações de máquinas propostas pela etapa precedente, até que uma designação factível seja encontrada.

O programa computacional correspondente ao método proposto foi escrito em linguagem C. Os resultados obtidos sobre exemplos da literatura são de boa qualidade, apesar do caráter não polinomial do problema tratado (WANG et al., 2023).

2. CONTEXTUALIZAÇÃO

WU et al. (2018), KULIK et al. (2021), SONG e CHENG (2022) e HADDADI e GATTAL (2022) propõem procedimentos de relaxação lagrangeana para resolução do problema da designação generalizada. Uma revisão sobre os métodos, algoritmos e avanços recentes da relaxação lagrangeana para resolução do problema da designação generalizada é realizada em BRAGIN (2024).

Os algoritmos genéticos são a base dos métodos desenvolvidos por ZHOU et al. (2020), CERGIPOZAN e TASAN (2022), e ZHI-BIN et al. (2022). ZHOU et al. (2022) desenvolveram dois algoritmos específicos para acelerar o desempenho e implementar ao mesmo tempo a computação paralela, enquanto que ZHI-BIN et al. (2022) combinam algoritmo genético e estratégia heurística “greedy” (AL-DOURI et al., 2021). CERGIPOZAN e TASAN (2022) apresentam o algoritmo e um estudo de caso de armazém de distribuição.

Dado caráter não polinomial do problema de designação generalizada (WANG et al., 2023), o uso de heurísticas e metaheurísticas é bastante comum, e pode ser encontrado em: D’AMBROSIO et al. (2020), EL YAFRANI et al. (2022), BOCCIA et al. (2023), FONTES et

al. (2023), FATHOLLAHI-FARD et al. (2023), LEE et al. (2023) e WANG et al. (2023). As metaheurísticas busca tabu e “simulated annealing” estão presentes em WANG et al. (2023) e FONTES et al. (2023). WANG et al. (2023) decompõem a resolução em três fases.

Uma revisão e avaliação de desempenho das heurísticas e metaheurísticas propostas na literatura para resolução do problema da designação é realizada em GEIBINGER et al. (2023). Um estudo dos diferentes modelos propostos para o problema da designação generalizada na literatura nos últimos 50 anos é proposto por PENTICO (2007).

O pacote computacional IBM-CPLEX é utilizado para resolver o modelo proposto em BILLING et al. (2020), MLADENOVIC et al. (2020), TRAN et al. (2022) e LEE et al. (2023). Os modelos da designação generalizada estudados em BOU SALEH et al. (2022) e BOCCIA et al. (2023) são resolvidos com auxílio do pacote computacional GUROBI.

3. PROBLEMA

Dadas as peças a fabricar, os roteiros de fabricação (sequências ordenadas de operações a executar sobre os lotes), as máquinas disponíveis e as durações das operações, trata-se de: 1) definir as máquinas para fabricar as peças entre aquelas do parque disponível; 2) designar as peças às máquinas definidas, respeitando-se as capacidades de fabricação.

Para a apresentação do método proposto, a seguinte notação será utilizada: a) nm_lt = número de lotes de peças a fabricar; b) $nm_pd[i]$ = número de peças a fabricar no lote i ; c) per_tr = período de trabalho; d) $nm_op[i]$ = número de operações executadas sobre o lote i ; e) $tipo[i, k = 1 \text{ a } nm_op[i]]$ = tipo de máquina utilizada para execução da k -ésima operação sobre o lote i ; f) $nm_mq[j]$ = número de máquinas disponíveis no tipo de máquina j ; g) $rv[j, m = 1 \text{ a } nm_mq[j]]$ = relação de velocidade da m -ésima máquina de tipo j , observando-se que as máquinas de referência dentro de cada tipo têm $rv[j, m]$ fixados iguais a 1; h) $duracao[i, k = 1 \text{ a } nm_op[i]]$ = duração da k -ésima operação executada sobre o lote i , com relação à máquina de referência do tipo. Assim, uma duração igual a 1.2 na máquina de referência corresponde a 0.6 numa máquina cuja relação de velocidade é o dobro. As TABELAS 1 e 2 fornecem os dados do exemplo tratado para ilustrar o procedimento.

TABELA 1 - Lotes, número de peças por lote, roteiros, duração das operações

Lote	$nm_pd[i]$	Roteiro	Duração
1	1	2 4 5 2	2.0 2.5 1.0 0.5
2	1	1 2 3 5	0.5 2.0 2.5 2.5
3	1	4 5	0.5 2.5

4	1	2 5	4.5 1.0
5	1	1 3 5	5.0 3.5 1.0
6	1	2 3 4	1.5 0.5 0.5
7	1	1 2 3 5	1.5 2.5 0.5 2.0

TABELA 2 - Tipos de máquinas, número de máquinas disponíveis, relação de velocidade

Tipo de máquina	nm_mq[j]	Relação de velocidade
1	2	1.0 1.3
2	3	1.0 1.0 1.2
3	3	1.0 1.2 1.5
4	2	1.0 1.3
5	2	1.0 1.0

4. MATRIZ DE CARGAS

Para realizar a designação, avalia-se primeiro a carga global de trabalho em relação a cada tipo de operação. A partir dos roteiros de fabricação dos lotes de peças, do número de peças a fabricar em cada lote e das durações das operações, calcula-se a matriz de cargas de trabalho [lotes de peças x tipos de máquinas]. Exemplo: TABELA 3. Esta matriz traz a carga de trabalho demandada pelos lotes de peças sobre cada tipo de máquina. Os elementos desta matriz são dados por:

$$\text{carga}[i, j] = \text{nm_pd}[i] \times \sum_{k | \text{tipo}[i, k] = j} \text{duração}[i, k]$$

Um lote de peças cujo roteiro de fabricação utiliza o trabalho de um mesmo tipo de máquina mais de uma vez tem sua carga de trabalho sobre o tipo de máquina igual à soma das durações destas operações multiplicadas pelo número de peças a fabricar nos respectivos lotes. Isto significa que estas operações são concentradas, num primeiro tratamento, num pacote como operação única.

Num segundo tratamento, estas operações retomam suas individualidades e podem ser executadas em máquinas diferentes, na medida em que os tipos e as máquinas disponíveis em cada tipo são conhecidos. Esta possibilidade deve ser considerada se, por exemplo, uma diminuição do tempo total de fabricação puder ser obtida através da execução de uma das operações em outra máquina disponível do mesmo tipo.

TABELA 3 - Matriz [lotes de peças x tipos de máquinas]

	tipo 1	tipo 2	tipo 3	tipo 4	tipo 5
lote 1	0.0	2.5	0.0	2.5	1.0
lote 2	0.5	2.0	2.5	0.0	2.5
lote 3	0.0	0.0	0.0	0.5	2.5
lote 4	0.0	4.5	0.0	1.0	0.0
lote 5	5.0	0.0	3.5	0.0	1.0
lote 6	0.0	1.5	0.5	0.5	0.0
lote 7	1.5	2.5	0.5	0.0	2.0

5. MODELO

Cada coluna da matriz [lotes de peças x tipos de máquinas], ou seja, cada tipo de máquina j , gera um problema específico a resolver: Dadas as máquinas disponíveis no tipo em questão, trata-se de encontrar uma combinação de máquinas capaz de efetuar as operações sobre os lotes que exigem sua intervenção e designar estas operações às máquinas desta combinação.

O modelo matemático em variáveis binárias (PM01j) fornecido abaixo permite a resolução do problema proposto.

PM01j

$$\begin{array}{ll} \text{minimizar} & \sum_{m=1}^{nm_mq[j]} \Phi \quad (\sum_{i=1}^{nm_lt} x_{im}) \times c[j, m] \end{array}$$

$$\begin{array}{ll} \text{sujeito a} & \sum_{i=1}^{nm_lt} \text{carga}[i, j] \times x_{im} \leq \text{cap}[j, m] \quad (m=1 \text{ a } nm_mq[j]) \quad (I) \end{array}$$

$$\begin{array}{ll} \sum_{m=1}^{nm_mq[j]} x_{im} = 1 & (i = 1 \text{ a } nm_lt) \quad (II) \end{array}$$

$$x_{im} = 0-1 \quad (i = 1 \text{ a } nm_lt, m= 1 \text{ a } nm_mq[j]) \quad (III)$$

$$\text{onde: } 1) \quad \Phi \left(\sum_{i=1}^{nm_lt} x_{im} \right) = \begin{cases} 1 & \text{se } \sum_{i=1}^{nm_lt} x_{im} > 0 \\ 0 & \text{caso contrário} \end{cases}$$

$$2) \quad \text{cap}[j, m] = \text{rv}[j, m] \times \text{per_tr} = \text{capacidade da máquina de tipo } j$$

$$3) \quad c[j, m] = \text{critério de otimização: a) número de máquinas; b) capacidade total das máquinas; c) custo de utilização das máquinas}$$

A função-objetivo minimiza um dos três critérios acima, à escolha do usuário. A 1ª restrição impede uma carga de trabalho maior que a capacidade das máquinas. A 2ª restrição estabelece que as operações podem ser designadas a uma única máquina. A 3ª restrição informa os valores possíveis da variável x_{im} .

6. MÉTODO

PM01j é NP-completo (WANG et al., 2023) e o número de problemas PM01j a resolver é igual ao número de tipos de máquinas utilizadas. Sejam: α o número de elementos carga[i, j] $\neq 0$ (para um j fixado, $i = 1$ a nm_lt) e β o número de máquinas de tipo j. O número de restrições de PM01j será igual a $\alpha + \beta$ (desconsiderando-se $x_{im} = 0-1$) e o número de variáveis será igual a $\alpha * \beta$. Se, para um certo tipo de máquina j, dispõe-se de 5 máquinas para executar 100 lotes de peças, PM01j apresentaria 105 restrições e 500 variáveis 0-1 a calcular. A determinação de um número tão grande de variáveis exige um tempo de cálculo inviável (HADDADI e GATTAL, 2022).

Propõe-se, então, a resolução de PM01j em duas fases: Na primeira fase, efetua-se uma definição de máquinas para o tipo considerado, o que permite assegurar o atendimento da carga global de trabalho. Isto permite reduzir o número β de máquinas a considerar e, por conseguinte, do número de variáveis $\alpha * \beta$. Na segunda fase, faz a designação dos lotes de peças às máquinas, resolvendo-se o problema por enumeração implícita ou heurísticas (DATE e NAGI, 2024). Cada coluna da matriz [lotes de peças x tipos de máquinas], ou seja, cada tipo de máquina j, gera um problema específico a resolver: Dadas as máquinas disponíveis no tipo em questão, trata-se de encontrar uma combinação de máquinas capaz de efetuar as operações

sobre os lotes que exigem sua intervenção e designar estas operações às máquinas desta combinação.

6.1 Definição das Máquinas

A definição das máquinas responsáveis pela fabricação das peças é feita separadamente para cada tipo de máquina j e independentemente do processo de designação dos lotes de peças às máquinas. Uma vez definidas as máquinas, tem-se um problema de designação de dimensão menor ou igual à dimensão do problema a resolver considerando-se todas as máquinas inicialmente propostas. A resolução do problema da mochila (FLESZAR, 2022) abaixo (MCHj), permite a realização desta definição. No modelo apresentado, a variável bivalente y_m designa a definição ($y_m = 1$) ou a não-definição ($y_m = 0$) da máquina m . A função-objetivo de MCHj minimiza o critério escolhido pelo operador (a , b ou c). A restrição principal (IV) assegura que a capacidade das máquinas disponíveis é suficiente para realizar a carga de trabalho globalmente demandada pelos lotes de peças.

MCHj

$$\begin{aligned}
 & \text{minimizar} && \sum_{m=1}^{nm_mq[j]} c[j, m] \times y_m \\
 & \text{sujeito a} && \sum_{m=1}^{nm_mq[j]} \text{cap}[j, m] \times y_m \geq \sum_{i=1}^{nm_lt} \text{carga}[i, j] \quad \text{(IV)} \\
 & && y_m = 0/1 \quad (m = 1 \text{ a } nm_mq[j]) \quad \text{(V)}
 \end{aligned}$$

A diminuição do número de máquinas inicialmente propostas leva a uma diminuição do coeficiente β e, por conseguinte, a uma diminuição do número de variáveis $\alpha \times \beta$ a determinar no problema de designação.

Propõe-se a resolução de MCHj de duas diferentes maneiras: 1) através da programação dinâmica (SOUZA et al., 2022), que permite a enumeração de todas as soluções factíveis para o problema tratado, inclusive a solução ótima; 2) através de um procedimento heurístico composto de três etapas: a) geração de combinações de máquinas; b) submissão das

combinações geradas à restrição principal de MCHj; c) classificação de algumas combinações factíveis em uma lista segundo seu valor de função-objetivo.

Como resultado deste procedimento obtém-se algumas ou todas as combinações de máquinas para as quais, na ordem do critério a otimizar, é verificada a existência de uma designação factível dos lotes de peças. O número de combinações estocadas no procedimento heurístico é um parâmetro definido pelo operador e seu valor foi fixado igual a 10 nos testes computacionais realizados. PM01j é NP-completo (WANG et al., 2023) e o número de problemas PM01j a resolver é igual ao número de tipos de máquinas utilizadas.

6.2 Designação das Operações às Máquinas

A resolução de MCHj na fase precedente gera as combinações de máquinas para as quais testa-se, nesta etapa, a existência de uma designação factível. Este teste é realizado primeiro para a combinação que apresenta o melhor valor de função-objetivo; em seguida – se a designação não é possível – para o segundo valor de função-objetivo, etc.

A simples resolução do sistema linear em variáveis bivalentes formado pelas restrições de PM01j por enumeração implícita (ou heurísticas) é uma solução possível para o problema da designação. Pode-se, também, definir um novo objetivo a atingir quando da resolução deste sistema, buscando-se assim uma designação otimizada dos lotes às máquinas e não uma designação qualquer. A função-objetivo de PM01j após a resolução de MCHj não é mais otimizável pois é igual a constante. A seguinte função-objetivo para o novo problema de designação (PMM01j) com as restrições de PM01j, é, então, proposta :

$$\begin{array}{llll} & nm_lt & nm_mq[j] & \\ \text{minimizar} & \sum_{i=1} & \sum_{m=1} & p[j, m] \times x_{im} \end{array} \quad (VI)$$

onde $p[j, m]$ é a penalidade atribuída pelo operador à utilização de cada máquina da combinação testada. Desta maneira, pode-se dirigir a designação dos lotes sobre certas máquinas (as mais rápidas ou as menos caras, etc.), à escolha do operador. Se o número de variáveis 0-1 a determinar for menor que 100, PMM01j é resolvido pelo algoritmo de enumeração implícita. Caso contrário, aplica-se o algoritmo “greedy” (AL-DOURI et al., 2021). Tal procedimento permite a obtenção de uma solução ótima ou realizável em um tempo de cálculo razoável.

O resultado obtido para o exemplo proposto é apresentado na TABELA 4. As penalidades para utilização das máquinas foram todas fixadas iguais a 1 e o período de trabalho igual a 10. O critério de otimização considerado foi a capacidade total das máquinas na fabricação.

TABELA 4 - Matriz [lotes de peças x máquinas]

máquina	1	2	3	4	5	6
tipo	1	2	2	3	4	5
número	1	1	2	1	1	1
lote 1	0.0	2.5	0.0	0.0	2.5	1.0
lote 2	0.5	2.0	0.0	2.5	0.0	2.5
lote 3	0.0	0.0	0.0	0.0	0.5	2.5
lote 4	0.0	4.5	0.0	0.0	1.0	0.0
lote 5	5.0	0.0	0.0	3.5	0.0	1.0
lote 6	0.0	0.0	1.5	0.5	0.5	0.0
lote 7	1.5	0.0	2.5	0.5	0.0	2.0
carga	7.0	9.0	4.0	7.0	4.5	9.0

7. ALGORITMO

Os passos fundamentais do algoritmo desenvolvido para efetuar a definição e a designação das máquinas estão resumidos no algoritmo d&d abaixo. Neste algoritmo está prevista a interferência do operador de modo a levar em conta o conhecimento específico do mesmo sobre as particularidades do problema tratado em detrimento das técnicas matemáticas de otimização.

No caso de o usuário do programa desejar promover alterações na definição e designação das máquinas obtidas, o programa apenas verifica a factibilidade das modificações indicadas, ou seja, se a capacidade das máquinas não é excedida, e apresenta o resultado desta verificação. Em vista desta informação, o usuário pode desistir da alteração ou implementá-la mesmo no caso de capacidade insuficiente.

O problema da mochila MCH_j e o problema da designação PMM01_j são gerados automaticamente pelo programa a partir dos dados iniciais e resolvidos por meio de uma das duas versões algorítmicas disponíveis para cada caso, levando-se em conta o número de variáveis bivalente: se o número destas é inferior a 100, utiliza-se a programação dinâmica e a enumeração implícita, respectivamente. Caso contrário, utiliza-se procedimentos heurísticos.

algoritmo d&d

dados: roteiros e tempos, penalidades das máquinas e critério de otimização

- 1) obtenção da matriz de cargas [lotes de peças x máquinas]
- 2) resolução do problema da mochila MCH j e do problema da designação PMM01j
- 3) apresentação do resultado obtido ao operador
- 4) possibilidade de efetuar modificações sobre a solução apresentada

fim algoritmo d&d**8. TESTES**

A TABELA 5, abaixo, exhibe o resultado dos testes computacionais realizados sobre exemplos da literatura. Na primeira coluna desta tabela é fornecida a referência do problema escolhido; na segunda é dado o par $[r \times s]$, onde r é o número de peças e s é o número de tipos de máquinas; na terceira e quarta colunas são fornecidos, respectivamente, o número de máquinas disponíveis para o planejamento (u) e o número de operações a executar (v). Nas colunas seguintes, T_{pd} é o tempo de cálculo em segundos que o programa despende para efetuar a definição das máquinas com a ajuda da programação dinâmica e T_{eh} com o algoritmo de enumeração heurística, T_{ei} é o tempo de cálculo para o algoritmo de enumeração implícita e T_{gu} para o algoritmo “greedy”. Na Tabela 5, o caractere (-) indica que o tempo computacional foi inferior a 0.1 segundo e o caractere (>) indica um tempo superior a 30 minutos.

TABELA 5 - Exemplos e Tempos de Cálculo em Segundos (Pentium, 4,6 GHz)

Exemplo	$[r \times s]$	u	v	T_{pd}	T_{eh}	T_{ei}	T_{gu}
KUMAR et al., 1986	7 x 5	12	21	-	-	-	-
KUMAR et al., 1986	9 x 7	14	45	-	-	1,3	-
KUMAR et al., 1986	9 x 7	21	45	-	-	9,8	-
CHAN e MILNER, 1981	10 x 15	15	46	-	-	-	-
SRINIVASAN et al., 1990	20 x 10	10	49	-	-	-	-
HARHAKAKIS et al., 1990	20 x 20	20	79	-	-	-	-
BURBIDGE, 1975	43 x 16	16	127	-	-	-	-
BURBIDGE, 1975	43 x 16	48	127	-	-	>	-

9. CONSIDERAÇÕES FINAIS

O procedimento de resolução em duas etapas de PM01j (busca de soluções factíveis por programação dinâmica ou enumeração de um conjunto de soluções factíveis, seguida de enumeração implícita ou algoritmo “greedy”) explora as particularidades do problema tratado (definição de máquinas, designação dos lotes de peças sobre as máquinas) para obter uma solução ótima ou factível em tempo de cálculo aceitável ou irrelevante sobre microcomputador (Pentium, 4,6 GHz).

A TABELA 5 apresenta a resolução de 8 exemplos extraídos da literatura. Os resultados foram os mesmos pelos 4 algoritmos disponíveis. O tempo computacional exigido pelo algoritmo de enumeração implícita é o mais elevado, conforme esperado. Deve-se observar na TABELA 5 que quando o número de máquinas disponíveis para o planejamento é igual ao número de tipos de máquinas, o tempo de cálculo do algoritmo de enumeração implícita é irrelevante, isto porque o programa dispõe de uma heurística para verificação da existência de designação factível, quando existe apenas uma máquina no tipo. Caso a heurística encontre uma resposta afirmativa, a designação encontrada é confirmada como solução.

A programação dinâmica mostrou-se tão rápida quanto a enumeração heurística de 10 soluções factíveis. O inconveniente da programação dinâmica, todavia, encontra-se na ocupação de memória da máquina, pois é necessário armazenar as diferentes soluções ótimas em cada estágio para se efetuar a recuperação da trajetória na etapa subsequente. Isto limita a sua utilização para resolução de problemas de grande porte.

O programa correspondente ao método proposto foi escrito em linguagem C e testado utilizando-se microcomputador. Ele gera automaticamente os programas matemáticos a resolver e apresenta ao operador a definição e a designação obtida. Tendo em vista a facilidade com que tal ferramenta de cálculo gera alternativas de decisão e sua proximidade com a realidade industrial (máquinas de velocidade variável, durações operatórias variáveis, capacidades das máquinas, etc.) pode-se imaginar sua utilização em tempo real.

REFERÊNCIAS

AL-DOURI, T.; HIFI, M.; ZISSIMOPOULOS, V. **An iterative algorithm for the max-min knapsack with multiple scenarios**. Operational Research, 21, p. 1355-1392, 2021. <https://doi.org/10.1007/s12351-019-00463-7>

BILLING, C.; JAEHN, F.; WENSING, T. **Fair task allocation problem**. Annals of Operations Research, 284, p. 131-146, 2020. <https://doi.org/10.1007/s10479-018-3052-3>

BOCCIA, M.; MASONE, A.; STERLE, C.; MURINO, T. **The parallel AGV scheduling problem with battery constraints: A new formulation and a matheuristic approach**. European Journal of Operational Research, 307, 2, p. 590-603, 2023. <https://doi.org/10.1016/j.ejor.2022.10.023>

BOU SALEH, M.; GRUNDER, O.; EL ASSANI, A. H. **Mixed-integer linear programming for specialized education and home care services**. IFAC-PapersOnLine, 55, 10, p. 3130-3135, 2022. <https://doi.org/10.1016/j.ifacol.2022.10.210>

BRAGIN, M.A. **Survey on Lagrangian relaxation for MILP: importance, challenges, historical review, recent advancements, and opportunities**. Annals of Operations Research, 333, p. 29-45, 2024. <https://doi.org/10.1007/s10479-023-05499-9>

BURBIDGE, J. L. **The Introduction of Group Technology**. John Wiley, 1975.

CACCHIANI, V.; IORI, M.; LOCATELLI, A.; MARTELLO, S. **Knapsack problems - An overview of recent advances. Part I: Single knapsack problems**. Computers & Operations Research, 143, p. 105692, 2022. <https://doi.org/10.1016/j.cor.2021.105693>

CERGIBOZAN, Ç.; TASAN, A. S. **Genetic algorithm based approaches to solve the order batching problem and a case study in a distribution center**. Journal of Intelligent Manufacturing, 33, p. 137-149, 2022. <https://doi.org/10.1007/s10845-020-01653-3>

CHAN, H. M.; MILNER, D. A. **Direct Clustering Algorithm for group formation in cellular manufacture**. Journal of Manufacturing Systems, 1, p. 399-416, 1981. [http://dx.doi.org/10.1016/S0278-6125\(82\)80068-X](http://dx.doi.org/10.1016/S0278-6125(82)80068-X)

D'AMBROSIO, C.; MARTELLO, S.; MONACI, M. **Lower and upper bounds for the non-linear GAP**. Computers & Operations Research, 120, 104933, 2020. <https://doi.org/10.1016/j.cor.2020.104933>

DATE, K.; NAGI, R. **Optimal and heuristic solutions for placing multiple finite-size rectangular facilities in an existing layout**. International Journal of Production Research, 62, p. 24-44, 2024. <https://doi.org/10.1080/00207543.2023.2284203>

EL YAFRANI, M.; SUNG, I.; KRACH, B.; KATSILIERIS, F.; NIELSEN, P. **Analysis of a local search heuristic for the GAP with resource-independent task profits and identical resource capacity**. Engineering Optimization, 54(8), p. 1426-1440, 2022. <https://doi.org/10.1080/0305215X.2021.1940991>

FATHOLLAHI-FARD, A. M; WONG, K. Y.; ALJUAID, M. **An efficient adaptive large**

neighborhood search algorithm based on heuristics and reformulations for the GQA problem. Engineering Applications of Artificial Intelligence, 126, Part A, 106802, 2023. <https://doi.org/10.1016/j.engappai.2023.106802>

FLESZAR, K. A branch-and-bound algorithm for the quadratic multiple knapsack problem. European Journal of Operational Research, 298, 1 p. 89-98, 2022. <https://doi.org/10.1016/j.ejor.2021.06.018>

FONTES, D. B. M. M.; HOMAYOUNI, S. M.; GONÇALVES, J. F. A hybrid particle swarm optimization and simulated annealing algorithm for the job shop scheduling problem with transport resources. European Journal of Operational Research, 306, 3, p. 1140-1157, 2023. <https://doi.org/10.1016/j.ejor.2022.09.006>

GEIBINGER, T.; KLETZANDER, L.; MUSLIU, N. Instance space analysis for the generalized assignment problem. In: Lecture Notes in Computer Science, vol 13838, 2023. https://doi.org/10.1007/978-3-031-26504-4_30

HADDADI, S.; GATTAL, E. Combining data reduction, MIP solver and iterated local search for generalized assignment. International Journal of Management Science and Engineering Management, 17, 2, 2022. <https://doi.org/10.1080/17509653.2021.1970039>

HARHALAKIS, G.; NAGI, R.; PROTH, J. M. An efficient heuristic in manufacturing cell formation for GT Applications. International Journal of Production Research, 28, 1, p. 185-198, 1990. <https://doi.org/10.1080/00207549008942692>

KULIK, A.; SHACHNAI, H.; TAMIR, G. On lagrangian relaxation for constrained maximization and reoptimization problems. Discrete Applied Mathematics, 296, p. 164-178, 2021. <https://doi.org/10.1016/j.dam.2020.10.001>

KUMAR, K. R.; KUSIAK, A.; VANELLI, A. Grouping of parts and components in FMS. European Journal of Operational Research, 24, 3, p. 387-397, 1986. [https://doi.org/10.1016/0377-2217\(86\)90032-9](https://doi.org/10.1016/0377-2217(86)90032-9)

LEE, J.; KIM, B.; KIM, S. H. A metheuristic algorithm for block assignment problem in long-term production planning in the shipbuilding industry. Computers & Industrial Engineering, 184, 109603, 2023. <https://doi.org/10.1016/j.cie.2023.109603>

MANIEZZO, V.; BOSCHETTI, M. A., STÜTZLE, T. The generalized assignment problem. In: Matheuristics. EURO Advanced Tutorials on OR. Springer, 2021. https://doi.org/10.1007/978-3-030-70277-9_1

MLADENOVIĆ, M.; DELOT, T.; LAPORTE, G.; WILBAUT, C. The parking allocation problem for connected vehicles. Journal of Heuristics, 26, p. 377-399, 2020. <https://doi.org/10.1007/s10732-017-9364-7>

PENTICO, D. W. **Assignment problems: A golden anniversary survey.** European Journal of Operational Research, 176, 2, p. 774-793, 2007. <https://doi.org/10.1016/j.ejor.2005.09.014>

SONG, M.; CHENG, L. **Solving the reliability-oriented generalized assignment problem by Lagrangian relaxation and Alternating Direction Method of Multipliers.** Expert Systems with Applications, 205, 117644, 2022. <https://doi.org/10.1016/j.eswa.2022.117644>

SOUZA, E. A. G.; NAGANO, M. S.; ROLIM, G. A. **Dynamic programming algorithms and their applications in machine scheduling: a review.** Expert Systems with Applications, 190, 116180, 2022. <https://doi.org/10.1016/j.eswa.2021.116180>

SRINIVASAN, G.; NARENDAN, T. T.; MAHAVEDAN, B. **An assignment model for the part-families problem in GT.** International Journal of Production Research, 28, 1, p. 145-152, 1990. <https://doi.org/10.1080/00207549008942689>

TRAN, Q. N. H.; NGUYEN, N. Q.; AMODEO, L. YALAOUI, F.; CHEHADE, H. **Multidimensional assignment problem in the medical sector: a case study.** IFAC-PapersOnLine, 55, 10, p. 901-905, 2022. <https://doi.org/10.1016/j.ifacol.2022.09.416>

WANG, Y.; LIU, H.; PENG, B.; WANG, H.; PUNNEN, A. P. **A three-phase matheuristic algorithm for the multi-day task assignment problem.** Computers & Operations Research, 159, 106313, 2023. <https://doi.org/10.1016/j.cor.2023.106313>

WU, W.; IORI, M.; MARTELLO, S.; YAGIURA, M. **Exact and heuristic algorithms for the interval min-max regret generalized assignment problem.** Computers & Industrial Engineering, 125, p. 98-110, 2018. <https://doi.org/10.1016/j.cie.2018.08.007>

ZHI-BIN, H.; GUANG-TAO, F.; DAN-YANG, D.; CHEN, X.; ZHE-LUN, D.; ZHI-TAO, D. **Novel parallel hybrid genetic algorithms on the GPU for the generalized assignment problem.** Journal of Supercomputing, 78, p. 144-167, 2022. <https://doi.org/10.1007/s11227-021-03882-6>

ZHOU, Z.; Li, F.; ZHU, H.; XIE, H.; ABAWAJY, J. H.; CHOWDHURY, M. U. **An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments.** Neural Computing & Applications, 32, p. 1531-1541, 2020. <https://doi.org/10.1007/s00521-019-04119-7>