São José dos Campos - SP
LV SBPO
LV Simpósio Brasileiro de Pesquisa Operacional
São José dos Campos, SP - 6 a 9 de novembro de 2023

Configuração de sistemas de manufatura celular com múltiplos níveis

Luis Fernando Rodriguez Jimenez
Faculdade de Engenharia Elétrica-UNICAMP
Av. Albert Einstein 400, Campinas-SP
l229779@dac.unicamp.br

Washington Alves de Oliveira
Faculdade de Ciências Aplicadas-UNICAMP
Rua Pedro Zaccaria 1300, Limeira-SP
waoliv@unicamp.br

RESUMO

Para melhorar a competitividade no mercado, as empresas têm aumentado o interesse em atender e satisfazer a maior quantidade possível de clientes, fazendo com que as exigências por novos produtos aumentem cada dia mais. Deste modo, o volume e a demanda por produtos crescem e modificam continuamente. A configuração de empresas em formato de células de manufatura ajuda a empresa a se adaptar às mudanças do mercado, melhorando a flexibilidade e diminuindo o transporte desnecessário de material e estoques de produtos nos processos. Visando estudar e melhorar a implementação das células de manufatura, este trabalho tem o objetivo de propor métodos de resolução para o problema de *layout* de sistemas de manufatura celular de múltiplos níveis em ambientes dinâmicos. São propostas duas versões para os métodos de resolução *greedy randomized adaptive search procedure* e busca tabu, os quais foram codificados em linguagem *Python* e comparados numericamente utilizando exemplares sintéticos e encontrados na literatura.

PALAVRAS CHAVE. Sistema de manufatura celular, configuração em múltiplos níveis, metaheurísticas.

ABSTRACT

Companies have increased their interest in serving and satisfying the most significant number of customers to improve the market portion, then the demands for new products increase every day. In this way, the volume and demand for products grow and change continuously. The configuration of companies in manufacturing cells helps the company adapt to changes in the market, improving flexibility and decreasing the unnecessary transport of material and product stocks in the processes. To study, model, and improve manufacturing cells' implementation, this work proposes mathematical model and resolution methods for the multi-floor layout design model of a dynamic cellular manufacturing system. Two resolution methods are proposed, greedy randomized adaptive search procedure and tabu search coded in Python language and compared numerically using a series of instances.

KEYWORDS. Cellular manufacturing system, multi-floor layout design, metaheuristics

DOI: 10.59254/sbpo-2023-174858
http://live-proceedings.pantheonsite.io/p/174858

1. Introdução

Os sistemas de manufatura celular (CMS) têm surgido para melhorar a eficiência das oficinas de trabalho ao incluir mais flexibilidade nos fluxos de linha, permitindo maior adaptação às mudanças e misturas de produtos de volume médio causados pelo aumento da concorrência global e os ciclos de vida curto dos produtos [Ahkioon et al., 2009]. A manufatura celular (CM) envolve o processamento de produtos semelhantes em um cluster dedicado de máquinas ou processos de fabricação. A CM é uma aplicação da tecnologia de grupo (GT), que identifica a proximidade entre tipos de peças ou produtos para aproveitar suas similaridades na manufatura e design [Irani, 1999]. Essas células são formadas cosiderando as vantagens do GT, como tempos reduzidos de instalação, redução de estoque nos processos, melhor qualidade dos produtos, menor tempo de entrega, maior produtividade e melhor controle das operações. Wemmerlöv e Hyer [1986] descreveram que as etapas de design de um sistema de fabricação celular incluem: *Formação de célula* – agrupar produtos que têm operações semelhantes e máquinas que fazem processos semelhantes em uma célula (conjunto de máquinas); *Layout de grupo* – intracelular e intercelular; *Agendamento de tarefas em grupo* – programação dos processos para uma famílias de produtos; e *Alocação de recursos* – atribuição de ferramentas, recursos humanos, materiais e dispositivos de manuseio de materiais.

Ahkioon et al. [2009] apresentaram um modelo completo para o design de sistemas de manufatura celular, onde foi incorporado decisões de planejamento da produção e reconfiguração do sistema. Para ambientes dinâmicos, Kia et al. [2012] trabalharam com o layout do grupo para um sistema de manufatura celular com roteiros de processos alternativos, divisão de lotes e reconfiguração flexível. Um modelo de programação não linear inteiro misto foi apresentado e resolvido mediante o método de recozimento simulado. Atualmente, espaços são geralmente insuficientes e caros para a construção de uma fábrica na área urbana, exigindo construções realizadas na dimensão vertical da planta, ou seja, instalações em vários andares (níveis de execução dos processos de produção). Khaksar-Haghani et al. [2013] ilustraram por meio da Figura 1 este panorama de construção vertical para um sistema de fabricação celular. Note que a figura ilustra quatro níveis da fábrica, o acesso é por meio de um elevador (deslocamento de pessoas e produtos) em fluxo vertical, cada nível é composto por um conjunto de localizações (duas ou mais localizações podem formar uma célula), além disso cada nível tem fluxo horizontal de pessoas e produtos. Kia et al. [2014] apresentaram um modelo de programação inteiro misto para a configuração de sistemas de manufatura celular dinâmicos de múltiplos níveis, para o qual foi considerado simultaneamente a formação de células e o layout de grupo como sendo as decisões inter-relacionadas envolvidas no CMS. Soluções aproximadas de excelente qualidade foram obtidas a partir da aplicação de um algoritmo genético.

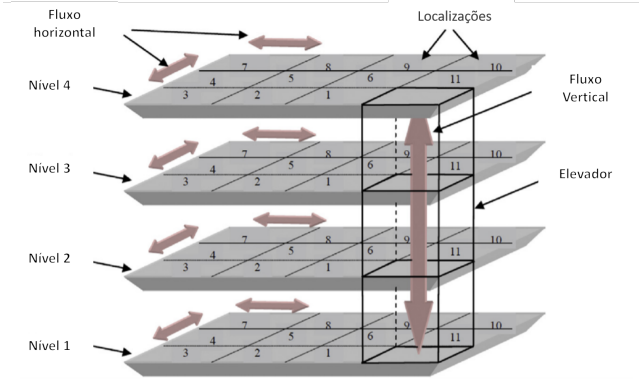

Figura 1: Configuração vertical em múltiplos níveis. Fonte: Khaksar-Haghani et al. [2013]

DOI: 10.59254/sbpo-2023-174858
http://live-proceedings.pantheonsite.io/p/174858

As contribuições deste artigo envolvem propor métodos de resolução para o problema do layout de sistemas de manufatura celular de múltiplos níveis em ambientes dinâmicos. A complexidade computacional da resolução de um modelo de programação justifica o uso de métodos heurísticos para se obter soluções aproximadas e de boa qualidade. Os métodos de resolução aplicados em exemplares do problema foram as metaheurísticas: *greedy randomized adaptive search procedure* (GRASP) e Busca Tabu (BT). Este trabalho inclui a implementação, aplicação e comparações numéricas entre os dois métodos de resolução propostos. Para validar a nossa abordagem, a linguagem *Python* foi usada para resolver um conjunto de exemplares do problema.

2. Descrição do problema

Kia et al. [2014] e Rodriguez [2021] apresentaram modelos matemáticos para o CMS de múltiplos níveis, os quais minimizam os custos totais do transporte de materiais intracelular (dentro de uma mesma célula), intercelular (entre células no mesmo nível) e entre células de diferentes níveis. Os modelos também consideram a compra de novas máquinas, a utilização, realocação e operações de processamento das máquinas. Na sequência apresentamos a descrição detalhada do problema integrado estudado para então explicar os métodos de resolução propostos nesse trabalho.

1. Cada tipo de produto possui uma sequência de operações para ser manufaturado. A demanda para cada tipo de produto em cada período, os recursos e tempos de processamento das operações de produtos para cada tipo de máquina são conhecidos e invariáveis no horizonte de planejamento. Uma operação de fabricação do produto é executada por apenas um tipo de máquina (existe pelo menos uma máquina capaz de processar essa operação) dentro da mesma célula ou em células diferentes (cópia da máquina). Não se tem estoque dos produtos, portanto, não se pode antecipar a fabricação de um produto para atender demandas futuras.

2. A capacidade limitada de cada tipo de máquina durante cada período de tempo é constante no horizonte de planejamento. Para atender aos requisitos de capacidade e reduzir/eliminar movimentos entre células, mais de uma cópia de cada tipo de máquina pode estar presente nas células. Considera-se que no primeiro período não existe máquina disponível para uso, então é necessário planejar a aquisição das máquinas para atender a demanda por produtos. A cada novo período, se a capacidade atual das máquinas não for suficiente para atender a demanda, outras máquinas são compradas e adicionadas ao conjunto daquelas máquinas já disponíveis. Em cada período em que existe capacidade excedente de produção na fábrica, no caso em que existe uma máquina ociosa, pode ser removida da célula e transferida para o depósito de máquinas, onde máquinas ociosas são mantidas para reduzir os custos suplementares destas máquinas. Além disso, permanecer com locais vazios na fábrica ajuda a tomar novas decisões para criação de novas células ou planejar novas operações de produtos. Sempre que for necessário aumentar a capacidade de tempo de processamento do sistema devido ao alto volume de demanda, as máquinas ociosas no depósito podem ser devolvidas às células. Cada tipo de máquina é multifuncional, ou seja, é capaz de executar pelo menos duas operações de fabricação sem impor custos de instalação. Assim, cada operação de fabricação de um tipo de produto pode ser processada em diferentes tipos de máquinas e com tempos diferentes de processamento.

3. O custo total com despesas gerais (suplementares, indiretos) de cada tipo de máquina, como manutenção, custo de energia e serviços gerais são conhecidos. Esse custo será considerado para cada máquina em cada período apenas se essa máquina for utilizada nas células para processar operações parciais. Portanto, as máquinas ociosas removidas das células e mantidas

DOI: 10.59254/sbpo-2023-174858
http://live-proceedings.pantheonsite.io/p/174858

no depósito de máquinas não resultam em despesas gerais adicionais. O custo variável de cada tipo de máquina é calculado com base na carga de trabalho atribuída à máquina. O custo de realocação de cada tipo de máquina é conhecido. Todos os tipos de máquinas podem ser movidos para o depósito de máquinas ou para qualquer local nas células. Esse custo é pago em algumas situações: a) ao transferir e instalar uma nova máquina comprada; b) ao realocar a máquina entre uma célula e o depósito de máquinas; e c) ao realocar uma máquina entre dois locais diferentes de uma mesma célula ou células diferentes. Por uma questão de simplicidade, supõe-se que o custo unitário da realocação de uma máquina seja o mesmo para todas as três situações. O custo de manuseio de material dos movimentos intracelular, intercelular e entre níveis estão fortemente relacionados com a distância percorrida.

4. A reconfiguração celular envolve: a) transferência das máquinas existentes entre diferentes locais de uma mesma célula ou células diferentes; b) compra e adição de novas máquinas às células; e c) transferência de máquinas entre células e depósito de máquinas devido à alteração dos requisitos de capacidade em períodos sucessivos. Os dispositivos de manuseio de material que movem os produtos são assumidos para transportar os produtos um a um.

5. Considera-se uma fábrica com vários níveis, para o qual cada nível é discretizado em vários locais com mesma dimensão. O número de locais em cada andar pode variar de uma fábrica para outra. Cada andar possui um ponto de chegada e um ponto de partida, onde as partes entram no piso no ponto de chegada e saem do piso a partir do ponto de partida. A localização do ponto de chegada e do ponto de partida em cada andar é conhecida. Existe um elevador que conecta todos os níveis e a localização deste elevador na fábrica é predeterminada. O layout em cada andar da fábrica tem múltiplos corredores (múltiplas linhas) e aquelas máquinas que ocupam quantidade igual de área de chão de fábrica deveria ser atribuídas a locais distintos, porém com as mesmas dimensões. Para transferir um produto de um andar para outro andar, primeiro o produto deve ser transferido da máquina que processa a operação do executor desse produto para o ponto de partida do piso atual por um dispositivo de transporte entre células. Em seguida, o produto deve ser transferido do piso atual para o piso de destino por um dispositivo de transporte vertical. Finalmente, um dispositivo de transporte entre células transfere o produto do ponto de chegada do piso de destino para a máquina que processa a próxima operação.

6. Um produto tem movimento intracelular entre máquinas de uma mesma célula quando precisa ser processado por operações consecutivas designadas nessas máquinas. Por outro lado, um movimento intercelular do produto para processar as operações ocorre entre máquinas de células diferentes, porém em um mesmo nível. Por fim, um movimento do produto para processar operações pode ocorrer entre níveis, sempre que as máquinas designadas para o processo não estão disponíveis em uma célula ou em diferentes células de um mesmo nível. O movimento intracelular, intercelular e entre níveis têm diferentes custos com base nos tipos de produto a ser fabricado. Assume-se que movimentos entre níveis realizados por dispositivos de transporte vertical são mais caros em comparação aos movimentos com dispositivos de transporte horizontal utilizados para o movimento intracelular ou intercelular no mesmo nível, e os movimentos intercelulares são mais custosos que os movimentos intracelulares.

7. O quantidade de células a serem formadas é especificado previamente. No entanto, o formato das células não é predeterminado e as células podem ser configuradas de maneira flexível em cada período do horizonte de planejamento. O limite máximo e mínimo do tamanho da célula

DOI: 10.59254/sbpo-2023-174858
http://live-proceedings.pantheonsite.io/p/174858

em termos de número de máquinas é conhecido à priori. Com o objetivo de monitorar melhor os processos de fabricação nas células, cada célula deve ser formada em um mesmo nível e as máquinas atribuídas a uma célula não podem ser localizadas em níveis diferentes.

3. Métodos de Resolução

3.1. Método GRASP

O método *greedy randomized adaptive search procedure* (GRASP) é uma técnica de busca adaptativa gananciosa com alto grau de aleatoriedade, onde cada iteração fornece uma solução para o problema a ser resolvido. O método GRASP tem sido aplicado a uma ampla gama de problemas de otimização combinatória, desde programação e roteamento até desenho e balanceamento de turbinas [Festa e Resende, 2009]. Cada iteração do GRASP consiste de duas fases: de construção, que cria de forma inteligente uma solução factível por meio de uma função gananciosa aleatória adaptativa; e de busca local, que otimiza/melhora a solução construída na primeira fase.

3.1.1. Configuração dos parâmetros para o método GRASP

– *Função índice para ordenar a escolha de designação de máquinas em locais*: A capacidade T_m de cada cópia de máquina é usada como critério de otimização para a designação de cada máquina m em um local disponível l. A compra de novas máquinas afeta a capacidade de produção.
– *Lista de candidatos para designação de máquinas em locais (LCl)*: Lista contendo todos os tipos de máquinas ordenadas de forma decrescente mediante a função índice de capacidade de máquina.
– *Lista de candidatos para designação de máquinas em células (LCc)*: Lista contendo todos as possíveis duplas de máquinas (designadas em locais vizinhos na célula) ordenadas de acordo com a soma total das operações designadas nelas.
– *Lista restrita de candidatos para designação de máquinas em locais (RLCl)*: Essa lista tem tamanho fixo p durante todo o processo de construção de uma solução factível, entra na lista as máquinas da LCl com maior função índice.
– *Lista restrita de candidatos para designação de máquinas em células (RLCc)*: Essa lista tem tamanho fixo p durante todo o processo de construção de uma solução factível, entra na lista as duplas de máquinas da LCc com maior soma total de operações.
– *Critério de desistência de uma configuração celular*: Se o algoritmo não completar a configuração celular durante um número predefinido de iterações, reinicia a construção de uma nova solução.
– *Critério de parada local*: O algoritmo tenta melhorar a solução inicial construída durante um número predefinido de iterações, então reinicia a fase de construção de uma nova solução.
– *Critério de parada global*: A construção de uma solução inicial e o processo de busca local formam uma iteração global do método. O ciclo encerra depois de um número predefinido de iterações globais. A probabilidade de encontrar melhores soluções aumenta conforme o número iterações globais, o qual define a quantidade de soluções geradas.

As Rotinas 1 e 2 descrevem o passo a passo de cada da fase do método GRASP proposto.

Rotina 1: Fase de construção GRASP
Entrada: Todos os elementos candidatos que são partes de uma possível solução
Saída: Solução factível
Passo 1. (*Designar operações às máquinas*) Em cada período, enquanto houver operações na lista de operações a serem designadas a uma cópia de máquina disponível, repita:
Considere na seguinte sequência os conjuntos i) de máquinas disponíveis na fábrica, ii) de máquinas no depósito de máquinas e iii) de máquinas que podem ser compradas. Selecionar aleatoriamente uma operação e uma máquina, se tem capacidade e pode processar a operação, então a operação é

DOI: 10.59254/sbpo-2023-174858
http://live-proceedings.pantheonsite.io/p/174858

designada para essa máquina. Caso contrário, refaça a mesma verificação para a próxima máquina. Se uma máquina no conjunto de máquinas disponíveis na fábrica permanecer ociosa, é movida para o depósito de máquinas.
Passo 2. (*Listas de candidatos*) Criar a lista LCl – lista de máquinas para a qual as operações foram designadas no **Passo 1**. Essa lista é ordenada a partir do valor das capacidades dessas máquinas. Criar a lista RLCl – lista contendo as primeiras posições (máquinas) da lista LCl.
Passo 3. (*Designar máquinas aos locais*) Enquanto houver máquinas na RLCl, repita:
Selecionar aleatoriamente máquina da RLCl e alocar em algum local vazio. Atualizar LCl e RLCl.
Passo 4. (*Listas de candidatos*) Criar a lista LCc (lista de duplas vizinhas de máquinas) ordenada a partir do valor da soma das operações que foram designadas para as máquinas que fazem parte da dupla. Criar a lista RLCc – lista contendo as primeiras posições (duplas de máquinas) da lista LCc.
Passo 5. (*Designar locais (com máquinas) para formar as células*) Enquanto existir máquinas não alocadas em uma célula ou não seja atingido o critério de parada (número de iterações), repita:
Passo 5.1. Selecionar aleatoriamente uma dupla da RLCc e alocar essa dupla de máquinas em alguma célula que ainda não atingiu o seu tamanho máximo.
Passo 5.2. Atualizar LCc e RLCc considerando que: i) as duplas são redefinidas de acordo com as vizinhanças; ii) uma máquina pode ficar sem uma dupla, assim permanece na fila de entrada pois é vizinha somente de uma máquina que já foi alocada na célula.
Passo 5.3. Para cada célula cujo tamanho máximo ainda não foi atingido, alocar uma das máquinas sem dupla que é vizinha de pelo menos uma das máquinas que já pertence a essa célula.

Rotina 2: Fase de busca local GRASP
Entrada: Solução factível (S)
Saída: Solução melhorada
Passo 1. Enquanto o número máximo de iterações não é atingido, repita:
Selecionar aleatoriamente dois locais l_f e l'_f não vazios e trocar as máquinas alocadas neles. Obter a solução $\bar{S}$ resultante dessa troca. Se $f(\bar{S}) < f(S)$, então atualize a solução, $S \leftarrow \bar{S}$.

3.2. Método da Busca Tabu

Busca Tabu (BT) é um método desenvolvido para resolver problemas de otimização combinatória. De forma geral, BT é um procedimento adaptativo com a capacidade de fazer uso de muitos outros métodos, como algoritmos de programação linear e heurísticas especializadas, direcionado para superar as limitações da otimização local [Glover, 1989]. Uma característica importante é que pode ser sobreposto sobre outros procedimentos para evitar pontos estacionários [Glover, 1990]. Na sequência é detalhada a configuração dos parâmetros para o método BT proposto.

3.2.1. Configuração dos parâmetros para o método da Busca Tabu

– *Solução inicial*: A solução inicial é criada aleatoriamente de forma similar à criação da solução factível na fase de construção do GRASP.
– *Lista de candidatos*: Os movimentos que serão avaliados em cada iteração são salvados na lista de candidatos. A quantidade de movimentos vai variar dependendo do tamanho do exemplo.
– *Lista Tabu de movimentos*: Os movimentos feitos em cada iteração estarão na lista Tabu até atingir um número de iterações predefinido. A quantidade de iterações muda dependendo do tamanho do exemplar resolvido. Esses movimentos podem sair da lista se um critério de aspiração é atingido.

DOI: 10.59254/sbpo-2023-174858
http://live-proceedings.pantheonsite.io/p/174858

– *Lista Tabu de operações*: Quando uma designação de operações em máquinas é realizada para criar a solução inicial, armazena-se na lista Tabu de operações, evitando repetir a mesma designação de operações em máquinas e ajudando a sair de ótimos locais. O número de iterações muda de acordo com o tamanho do exemplar.
– *Critério de aspiração*: Para que um movimento candidato que pertence a lista Tabu possa sair, é necessário que melhore a solução global atual. Caso contrário, o movimento vai continuar na lista Tabu enquanto uma determinada quantidade de iterações (tamanho da lista Tabu) ocorra.
– *Critério de parada local*: O critério de parada em busca local será quando cumprir com um número fixo de iterações, nesse caso o algoritmo vai criar uma nova solução inicial.
– *Critério de parada global*: O critério de parada global será quando atingir um número fixo de iterações definidas inicialmente.

A Rotina 3 resume o passo a passo do método BT proposto neste trabalho.

Rotina 3: Método da Busca Tabu
Entrada: Exemplar, CP_1, CP_2, CP_3 (três critérios de paradas), lista Tabu de operações vazia, lista Tabu de movimentos vazia, $n_1 = n_2 = n_3 = 1$, $A = \infty$.
Saída: Solução final
Passo 1. (*Solução inicial factível*) Construir uma solução inicial factível S_0 que não pertença a lista Tabu de operações, usando a **Rotina 1** da fase de construção do GRASP. Os **Passos 1–3** da **Rotina 1** são simplificados: as listas LCl e LCc não são ordenadas e as listas RLCl e RLCc são eliminadas; e os sorteios para construir a solução são realizados diretamente nas listas não ordenadas LCl e LCc. Se $f(S_0) < A$, então $A \leftarrow f(S_0)$. Atualizar a lista Tabu e selecionar um período aleatoriamente.
Passo 2. (*Melhoria de solução*) Enquanto $n_3 \leq CP_3$, selecionar um local l_f não vazio. Para cada outro local não vazio testar a troca de máquinas entre esses dois locais e verificar o impacto na solução atual S em relação ao valor da função objetivo. Salvar essas trocas de forma ordenada em relação a este impacto como uma nova lista de candidatos. Tornar a lista Tabu de movimentos vazia e selecionar o primeiro movimento desta nova lista de candidatos.
Passo 2.1. Se o movimento selecionado não está na lista Tabu de movimentos, ir para o **Passo 2.3**.
Passo 2.2. Se o movimento selecionado satisfaz o critério de aspiração, ir para o **Passo 2.3**. Caso contrário, selecionar o próximo movimento da lista de candidatos e ir para o **Passo 2.1**.
Passo 2.3. Realizar o movimento na solução atual S e obter uma nova solução $\bar{S}$. Inserir este movimento que foi realizado na lista Tabu de movimentos. Se $f(\bar{S}) < f(S_0)$, então $S_0 \leftarrow \bar{S}$. Atualizar a lista Tabu de movimentos, $S \leftarrow \bar{S}$, $n_3 \leftarrow n_3 + 1$ e ir para o **Passo 2**.
Passo 3. Se $n_2 \leq CP_2$, selecionar aleatoriamente um novo período, $n_2 \leftarrow n_2 + 1$ e ir para o **Passo 2**.
Passo 4. Se $f(S_0) < A$, então $A \leftarrow f(S_0)$ e $S_f \leftarrow S_0$. Se $n_1 \leq CP_1$, $n_1 \leftarrow n_1 + 1$, ir para o **Passo 1**. Caso contrário, Solução final é S_f.

---

4. Resultados computacionais

Para validar a qualidade dos algoritmos propostos, foram resolvidos 56 exemplares gerados sinteticamente. Para decidir a ordem de grandeza de cada parâmetro, 54 exemplares foram gerados tomando como base os valores de parâmetros disponibilizados por Kia et al. [2014]. Por outro lado, 02 exemplares foram retirados exatamente como está em Kia et al. [2014].

Os exemplares foram divididos em três classes: pequenos, medianos e grandes. As classes de exemplares são diferenciadas por distintas combinações dos parâmetros apresentados na primeira coluna da Tabela 1. A coluna de unidade de variação (u.v.) faz referência ao quanto está variando cada um dos parâmetros na escolha do seu valor para um exemplar. Por exemplo, o preço de compra

DOI: 10.59254/sbpo-2023-174858
http://live-proceedings.pantheonsite.io/p/174858

de máquina para os exemplares pequenos pode variar entre 3200 e 8000, sendo que a u.v. para este parâmetro é 100, o que significa que os valores podem aumentar ou diminuir em 100 unidades. A criação de cada exemplar é feita aleatoriamente utilizando os intervalos de valores ou valores fixados conforme a Tabela 1. Os algoritmos para os métodos de resolução propostos foram codificados na linguagem de programação *Python* e rodados em um notebook com processador Intel(R) Core(TM) i5-8300H 2.3GHz e 8GB RAM. Este capítulo apresenta os detalhes dos experimentos computacionais realizados e uma séria de estatísticas resumindo e comparando os resultados numéricos obtidos para o problema de layout para manufatura celular estudado.

Parâmetro	Pequeno	Mediano	Grande	u.v.
No. de períodos	2	3–4	5–8	1
No. de níveis	2	3–5	5–8	1
No. de locais por nível	5–6	6–8	9–10	1
No. de produtos	3–8	6–9	8–20	1
No. de op. por produto	2–5	3–8	9–12	1
No. de tipos de máquinas	3–8	6–10	11–15	1
No. de células por nível	2	2	2–3	1
Tempo de op. em máquinas	10–60	10–70	10–80	10
Custo intracelular	0,2–0,5	2–3,5	3–5	0,1
Custo intercelular	0,5–2	4–7	5–9	0,5
Custo interníveis	1–4	7–10	9–12	1
Custo fixo	160–500	300–600	500–800	10
Custo variável	0,1–0,5	0,1–0,5	0,1–0,5	0,1
Custo de realocação	60–100	70–150	80–180	10
Preço de compra de máquina	3200–8000	3500–5000	4000–9000	100
Demanda do produto	100–400	0–800	0–1200	50
Capacidade das máquinas	6000–10000	9000–58000	20000–80000	1000

Tabela 1: Parâmetros para construção dos exemplares

4.1. Descrição da resolução

Apresenta-se de forma detalhada uma solução para um exemplar pequeno com a finalidade de exemplificar melhor a sequência de informações apresentadas para o conjunto de soluções obtido como resultado dos testes numéricos. O Exemplar 6 consiste de 5 tipos de produtos com 3 operações distintas de processamento para cada um, cinco tipos de máquinas e dois períodos. Cada operação pode ser alternativamente processada em duas máquinas. A Tabela 2 descreve os dados relacionados com a capacidade da máquina (T_m) em tempo (segundos), custo operacional (α_m), custo variável (β_m), custo de realocação (δ_m) e preço de compra (PC_m) de cada tipo de máquina.

	T_m	α_m	β_m	δ_m	PC_m
M_1	7000	300	0,1	80	3500
M_2	8000	200	0,5	80	4000
M_3	9000	400	0,3	90	8000
M_4	10000	190	0,3	70	6000
M_5	10000	250	0,4	100	5000

Tabela 2: Informações das máquinas do Exemplar 6

A Tabela 3 apresenta os tempos de processamento de cada operação O para todos os tipos de produtos P em cada máquina M. Note na tabela que a operação 2 do produto 1 pode ser processada pelo tipo de máquina 3 (20 segundos de tempo de processamento) e pelo tipo de máquina 5 (30 segundos de tempo de processamento). A Tabela 4 descreve os custos de manuseio de material intracelular, intercelular e entre níveis de cada tipo de produto. Nessa mesma tabela também estão as demandas de cada tipo de produto para dois períodos. Finalmente, a Tabela 5 apresenta as distâncias entre os locais para cada um dos níveis. Neste exemplo, existem dois andares e cada

DOI: 10.59254/sbpo-2023-174858
http://live-proceedings.pantheonsite.io/p/174858

	P1			P2			P3			P4			P5		
	O1	O2	O3	O1	O2	O3	O1	O2	O3	O1	O2	O3	O1	O2	O3
M_1									20	30	20	20	10	30	
M_2			30			30	10	10			20				10
M_3		20		10	20			30		10			20		
M_4	20		20		20							30		20	
M_5	10	30		30		10	20		20						30

Tabela 3: Disponibilidade de operações das máquinas no Exemplar 6

Informação do produto	P1	P2	P3	P4	P5
Custo intracelular	0,3	0,4	0,5	0,2	0,2
Custo intercelular	0,8	1,2	1,8	0,5	1,5
Custo internível	4,0	3,0	4,0	2,0	2,0
Demanda ($t = 1$)	250,0	300,0	300,0	250,0	200,0
Demanda ($t = 2$)	350,0	250,0	300,0	300,0	100,0

Tabela 4: Informações dos produtos para o Exemplar 6

andar tem 5 locais. Presume-se que cada andar tem a mesma forma e que o ponto de chegada e de partida estejam no mesmo local (representado por A/D). As máquinas em cada piso devem ser agrupadas em duas células independentes, sendo os tamanhos inferior de 2 e superior de 3.

	L1	L2	L3	L4	L5	A/D
L1	0	1	2	1	2	1
L2	1	0	1	2	3	2
L3	2	1	0	1	2	3
L4	1	2	1	0	1	2
L5	2	3	2	1	0	1
A/D	1	2	3	2	1	0

Tabela 5: Distâncias entre localizações para os dois níveis no Exemplar 6

As Tabelas 6 e 7 e a Figura 2 detalham a solução do Exemplo 6. A Tabela 6 mostra a designação das operações de cada produto nos tipos de máquinas para cada período. No período 1, a operação 1 do produto 1 foi processada pela cópia 1 da máquina 4 (M_4, C_1), a operação 2 foi processada pela cópia 1 da máquina 3 (M_3, C_1) e finalmente a operação 3 volta para a cópia 1 da máquina 4 (M_4, C_1). Para o período 2, as operações 1, 2 e 3 do produto 1 foram processadas pela cópia 1 da máquina 5 (M_5, C_1), cópia 1 da máquina 3 (M_3, C_1) e cópia 2 da máquina 4 (M_4, C_1), respectivamente.

A Tabela 7 apresenta a alocação realizada de cópias de máquinas nos locais disponíveis. Por exemplo, a cópia 1 da máquina 3, cópia 2 da máquina 3, cópia 1 da máquina 5, cópia 2 da máquina 4 e a cópia 1 da máquina 4, são designados nos locais 1, 2, 3, 4 e 5 do nível 1. Para os períodos 1 e 2, o local 1 do nível 2 permaneceu vazio. Além disso, como no período 1 compra-se todas as máquinas utilizadas ($M_1, M_1, M_1, M_2, M_3, M_3, M_4, M_4, M_5$), não se tem máquinas no depósito. No período 2, utilizou-se as mesmas máquinas compradas no período 1 e não foram adquiridas novas máquinas, portanto o deposito também permaneceu vazio.

A Figura 2 ilustra as configurações das células de manufatura. Note que para o primeiro período, no nível 1, a célula 1 está constituída pelas máquinas M_5, C_1, M_4, C_2 e M_4, C_1, alocadas no locais L3, L4 e L5. Por outro lado, a célula 2 está constituída pelas máquinas M_3, C_1 e M_3, C_2 alocadas em L1 e L2. A rota do produto 1 no período 1 é descrita na sequência para exemplificar. A operação 1 do produto 1, representada por $P1, O1$, é processada na célula 1 pela cópia de máquina

DOI: 10.59254/sbpo-2023-174858
http://live-proceedings.pantheonsite.io/p/174858

T	P	O1	O2	O3
1	P1	M_4,C_1	M_3,C_1	M_4,C_1
	P2	M_3,C_1	M_3,C_2	M_5,C_1
	P3	M_2,C_1	M_2,C_1	M_1,C_1
	P4	M_3,C_2	M_1,C_2	M_4,C_2
	P5	M_1,C_2	M_1,C_3	M_5,C_1
2	P1	M_5,C_1	M_3,C_1	M_4,C_2
	P2	M_3,C_2	M_4,C_1	M_5,C_1
	P3	M_2,C_1	M_2,C_1	M_1,C_3
	P4	M_3,C_2	M_1,C_1	M_1,C_2
	P5	M_1,C_3	M_4,C_2	M_5,C_1

Tabela 6: Operações designadas e rotas dos produtos para o Exemplar 6

	Nível 1					Nível 2					Dep. máquinas	Máquinas compradas
Período	L1	L2	L3	L4	L5	L1	L2	L3	L4	L5		
1	M_3,C_1	M_3,C_2	M_5,C_1	M_4,C_2	M_4,C_1	Vazio	M_2,C_1	M_1,C_3	M_1,C_2	M_1,C_1	Vazio	$M_1,M_1,M_1,M_2,M_3,M_3,M_4,M_4,M_5$
2	M_5,C_1	M_4,C_1	M_3,C_1	M_1,C_2	M_1,C_3	Vazio	M_1,C_1	M_4,C_2	M_3,C_2	M_2,C_1	Vazio	Vazio

Tabela 7: Alocação das cópias das máquinas e máquinas adquiridas por período no Exemplar 6

M_4,C_1 alocada em L5 no nível 1. Então, o produto se move para a célula 2 até a cópia da máquina M_3,C_1 alocado em L1 no nível 1 para que seja processada a operação 2 $(P1,O2)$. Finalmente, o produto 1 processa a operação 3 $(P1,O3)$ novamente na célula 1, na cópia M_4,C_1 alocada em L5 no nível 1. Note que é acrescentado custos intercelular ao movimento do produto da célula 1 para a célula 2. Um custo entre níveis pode ser visto na rota do produto 5 no período 1, o qual é movimentado da célula 1 do nível 2 para a célula 2 do nível 2 (custo intercelular), e termina na célula 1 do nível 1 (custo internível). A solução tem custo total de 98430, incluindo os custos intracelular, intercelular e interníveis de 965, 2575 e 5980, respectivamente, 47500 de compra de máquinas, 5060 de custo geral de máquinas, 35150 de custo operacional de máquinas e 1200 de custo de realocação de máquinas.

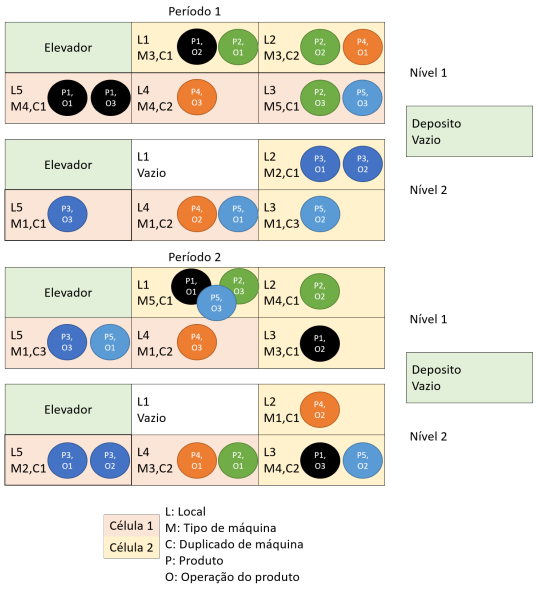

Figura 2: Configuração celular, alocação de operações e das máquinas para o Exemplar 6. Fonte: Própria

DOI: 10.59254/sbpo-2023-174858
http://live-proceedings.pantheonsite.io/p/174858

4.2. Resultados computacionais dos exemplares

A Tabela 8 apresenta os parâmetros de configuração do algoritmo para os exemplares pequenos, medianos e grandes, os quais são baseados nos parâmetros usados para resolver os Exemplares 1 e 2 de Kia et al. [2014]. O tempo limite para resolver cada exemplar foi de 1800 segundos.

	Pequenos		Medianos		Grandes	
	GRASP	Busca Tabu	GRASP	Busca Tabu	GRASP	Busca Tabu
Iterações globais	300	700	200	200	100	150
Iterações busca local	5000	3000	1500	2000	900	1000
Iterações designação de células	20	–	50	–	30	–
RLCl	7	–	10	–	10	–
RLCc	5	–	5	–	5	–
Lista Tabu operações	–	8	–	10	–	10
Lista Tabu locais	–	10	–	50	–	50
Tempo máximo (s)	1800	1800	1800	1800	1800	1800

Tabela 8: Parâmetros iniciais dos algoritmos para exemplares pequenos, medianos e grandes

A Tabela 9 compara os resultados obtidos entre os métodos para cada classe de exemplares. Nos exemplares pequenos, nota-se que cada exemplar 3, 4, 14 e 16 apresenta o mesmo valor de função objetivo nos dois métodos de resolução, o que pode sugerir que esses exemplares não possuem uma solução melhor. Note ainda que para 9 exemplares (1, 2, 6, 7, 8, 10, 15, 17 e 19), GRASP encontrou melhores soluções, enquanto em 5 exemplares (5, 11, 12, 13 e 18), Busca tabu foi vencedora. Em geral, GRASP apresenta melhores resultados para os exemplares pequenos. Adicionalmente, GRASP também performou melhor nos exemplares medianos, com exceção do exemplar 19, para o qual Busca Tabu tem custo menor. Finalmente, nos exemplares grandes, GRASP obteve melhores respostas em 12 exemplares (1, 2, 3, 4, 6, 7, 9, 12, 13, 15, 17 e 18), enquanto que Buscar Tabu foi melhor em 5 exemplares (5, 10, 11, 14 e 16) e, somente no exemplar 8, os dois métodos empataram.

	Pequenos				Medianos				Grandes			
Ex.	GRASP	Tempo	Busca Tabu	Tempo	GRASP	Tempo	Busca Tabu	Tempo	GRASP	Tempo	Busca Tabu	Tempo
1	**225450**	75	226350	51	**1200100**	381	1242470	140	**7821820**	892	8235200	1800
2	**149940**	62	155620	305	**1097545**	435	1214170	950	**14094395**	993	14514295	1340
3	**65775**	140	**65775**	201	**1439385**	993	1514260	1243	**17089200**	1112	17108300	1763
4	**77450**	189	**77450**	163	**2503065**	696	2639615	950	**40758810**	1456	41411520	1800
5	115925	70	**115355**	90	**3112360**	202	3286115	441	18044400	1800	**17895280**	1800
6	**101450**	75	108170	104	**2826375**	1800	3117025	1800	**23562190**	1569	24879370	1800
7	**91805**	37	92045	235	**4960765**	1800	6306520	1800	**30727440**	1800	30858950	1750
8	**135100**	984	138695	632	**3412730**	980	3775110	343	**42606670**	1800	**42606670**	1800
9	**111165**	301	**111165**	628	**1985045**	842	2260825	1080	**28557595**	1800	28989130	1663
10	**148005**	582	165980	359	**1504625**	880	1558635	1800	18777810	1800	**18329895**	1800
11	134545	453	**128805**	1800	**3124705**	1800	3194720	1800	21161635	1800	**20546090**	1800
12	113645	51	**108625**	1800	**3150140**	1800	3350285	1800	**25765565**	1800	26129995	1800
13	146005	80	**137895**	800	**2813420**	275	2950070	1800	**24365560**	917	24577085	1569
14	**139855**	720	**139855**	398	**3115000**	395	3220435	1800	17660385	1556	**16817760**	1775
15	**123235**	511	135995	1800	**2596685**	447	2701875	1800	**19927175**	1491	20240165	1800
16	**102060**	304	**102060**	503	**6971550**	564	7717410	1800	51923230	1369	**51678900**	1800
17	**97165**	76	98830	190	**4659945**	1800	5217697	525	**48201560**	1800	49608945	1800
18	123650	73	**113035**	140	**3377535**	1800	3910625	1800	**39763995**	1127	39820520	1800
19	**92200**	80	92320	650	202195	1700	**198610**	1451	–	–	–	–

Tabela 9: Comparação do valor da função objetivo dos métodos de resolução propostos

5. Conclusão e trabalhos futuros

A principal contribuição deste trabalho foi o desenvolvimento de dois métodos metaheurísticos (GRASP e Busca Tabu) para resolver um problema de difícil solução. Para validar a qualidade dos métodos propostas foram resolvidos três classes de exemplares totalizando 56 exemplares. Estatística descritiva foi usada para comparar os resultados obtidos. Em relação ao valor da função objetivo e tempo computacional, o método GRASP obteve melhor desempenho na maioria dos exemplares em todas as classes. No entanto, a Busca Tabu obteve destaque em alguns casos. Em

DOI: 10.59254/sbpo-2023-174858
http://live-proceedings.pantheonsite.io/p/174858

geral, os métodos de resolução propostos são promissores, exibindo comportamento satisfatório enquanto determina soluções factíveis e de boa qualidade em tempo computacional aceitável. Como trabalho futuro, as boas características do método GRASP proposto serão incorporadas na Busca Tabu, derivando um método híbrido com a finalidade de melhorar o desempenho do último. Além disso, pode ser interessante potencializar a segunda fase do método GRASP utilizando Busca Tabu. Adicionalmente, visando avaliar a qualidade dos métodos propostos, será desenvolvido um modelo de programação matemática para ser resolvido em um solver comercial em exemplares de dimensão moderada. Devido a dificuldade encontrada na geração do conjunto de exemplares para o problema em estudo, seria interessante desenvolver um algoritmo gerador de exemplares, para o qual precise apenas introduzir as grandezas dos parâmetros exigidos, facilitando na implementação de soluções para problemas gerais relacionados com sistemas de manufatura celular configurados em múltiplos níveis.

Agradecimentos

Este trabalho foi parcialmente financiado pelo CNPq (processo 309925/2021-5) e FAPESP (processo 2020/09838-0).

Referências

Ahkioon, S., Bulgak, A. A., e Bektas, T. (2009). Integrated cellular manufacturing systems design with production planning and dynamic system reconfiguration. *European journal of operational research*, 192(2):414–428.

Festa, P. e Resende, M. G. (2009). An annotated bibliography of grasp–part i: Algorithms. *International Transactions in Operational Research*, 16(1):1–24.

Glover, F. (1989). Tabu search – Part I. *ORSA Journal on computing*, 1(3):190–206.

Glover, F. (1990). Tabu search: A tutorial. *Interfaces*, 20(4):74–94.

Irani, S. A. (1999). *Handbook of cellular manufacturing systems*. John Wiley & Sons.

Khaksar-Haghani, F., Kia, R., Mahdavi, I., e Kazemi, M. (2013). A genetic algorithm for solving a multi-floor layout design model of a cellular manufacturing system with alternative process routings and flexible configuration. *The International Journal of Advanced Manufacturing Technology*, 66(5-8):845–865.

Kia, R., Khaksar-Haghani, F., Javadian, N., e Tavakkoli-Moghaddam, R. (2014). Solving a multi-floor layout design model of a dynamic cellular manufacturing system by an efficient genetic algorithm. *Journal of Manufacturing Systems*, 33(1):218–232.

Kia, R., Baboli, A., Javadian, N., Tavakkoli-Moghaddam, R., Kazemi, M., e Khorrami, J. (2012). Solving a group layout design model of a dynamic cellular manufacturing system with alternative process routings, lot splitting and flexible reconfiguration by simulated annealing. *Computers & operations research*, 39(11):2642–2658.

Rodriguez, F. (2021). Modelagem e resolução de um problema para desenhar a disposição de um sistema de manufatura celular com múltiplos níveis. Dissertação (mestrado), Universidade Estadual de Campinas, Faculdade de Ciências Aplicadas, Limeira, SP. URL `https://hdl.handle.net/20.500.12733/3710`.

Wemmerlöv, U. e Hyer, N. L. (1986). Procedures for the part family/machine group identification problem in cellular manufacturing. *Journal of Operations Management*, 6(2):125–147.

DOI: 10.59254/sbpo-2023-174858
http://live-proceedings.pantheonsite.io/p/174858