

RESOLUÇÃO DO PROBLEMA DE ALOCAÇÃO E ROTEAMENTO DE OPERADORES EM CÉLULAS DE MANUFATURA: ABORDAGEM EXATA E HEURÍSTICA

Victor Mouffron Carvalho Machado

Universidade Federal Fluminense - UFF
Rua Passo da Pátria, 156, Bloco Computação, São Domingos - Niterói - RJ - CEP: 24210-240
victormachado@id.uff.br

Tiago Araújo Neves

Universidade Federal Fluminense - UFF
Av. dos Trabalhadores, 420, Vila Santa Cecília - Volta Redonda, RJ - Brasil - 27255-125
tneves@ic.uff.br

Luiz Satoru Ochi

Universidade Federal Fluminense - UFF
Rua Passo da Pátria, 156, Bloco Computação, São Domingos - Niterói - RJ - CEP: 24210-240
satoru@ic.uff.br

RESUMO

O Problema de Alocação e Roteamento de Operadores em Células de Manufatura (PAROCM) é um problema encontrado no ambiente industrial e pouco explorado na literatura. Este problema consiste em minimizar o número de operadores necessários para produzir uma determinada quantidade de produtos, definir quais operações cada operador realizará e a rota que seguirá em seu ciclo de trabalho. Este trabalho desenvolve uma formulação de programação inteira para resolução do PAROCM e uma heurística simples. A resolução das instâncias geradas mostrou bons resultados, apesar da sensibilidade do modelo ao *takt-time* da célula. O aluno de iniciação científica foi responsável por redigir o texto, modelar o problema, propor o algoritmo heurístico, gerar as instâncias a partir da literatura, implementar os códigos e executar os testes.

PALAVRAS CHAVE. Células de manufatura, PAROCM, Programação Inteira, Heurística.

Área principal: Programação Matemática, Otimização Combinatória, Metaheurísticas.

ABSTRACT

The Manufacturing Cell Operators Allocation and Routing Problem (MCOARP) is commonly found in the industrial environment. However, it seems scarcely addressed in the literature. This problem consists in minimizing the numbers of operators needed to produce a certain amount of products, defining the operations that they will work on and their routes. This paper proposes an integer programming formulations to solve MCOARP, as well a simple heuristic. The solution of the proposed instances showed good results, in spite of the model sensitivity to the cell takt-time. The scientific initiation student was responsible for redacting the paper, modelling the problem, proposing the heuristic, generating the instances, implementing the code and running the tests.

KEYWORDS. Manufacturing Cell, MCOARP, Integer Programming, Heuristic.

Main area: Mathematical Programming, Combinatorial Optimization, Metaheuristics.

1. Introdução

A competitividade na indústria gera pressões pela redução de custos e pelo o aumento da produtividade. Para atingir esses objetivos, alterações no produto e no processo de produção são comumente realizadas. Uma alternativa é a utilização otimizada dos recursos, que pode gerar ganhos significativos, sem a necessidade de grandes investimentos em melhorias e longos prazos para aplicação.

O Problema de Alocação e Roteamento de Operadores em Células de Manufatura (PAROCM) se encaixa perfeitamente neste contexto. Qualquer solução que não minimize o número de operadores necessários a uma produção mensal causa um aumento no custo de produção e, ainda, pode gerar uma redução da produtividade, devido à presença de recursos desnecessários à produção do volume requisitado pelo cliente.

Portanto, surge a necessidade de determinar o número mínimo de operadores necessários para produzir um dado volume de produtos. Modelos de programação matemática podem ser utilizados para resolver este problema, assim como feito em outros problemas relacionados à indústria.

Métodos heurísticos não garantem encontrar soluções ótimas, contudo são boas alternativas quando são constatadas dificuldades na aplicação de métodos exatos, como, por exemplo, elevado tempo computacional e consumo exacerbado de memória. O PAROCM é NP-difícil, pois é uma generalização do Problema de Localização e Roteamento que é NP-Difícil [Nagy (2007)]. Por isso, optou-se por utilizar também uma abordagem heurística.

O restante deste trabalho encontra-se organizado da seguinte forma: a Seção 2 descreve detalhadamente o problema e o seu contexto. A Seção 3 apresenta a formulação proposta neste trabalho. A Seção 4 apresenta a heurística utilizada. A Seção 5 expõe alguns aspectos sobre as instâncias do problema utilizadas para os testes computacionais, bem como os resultados obtidos. Por último, a Seção 6 conclui o trabalho e apresenta sugestões para futuros temas relacionados ao problema.

2. O PAROCM

As células de manufatura são uma forma de layout desenvolvido pela Toyota Motor Company e se caracterizam por uma configuração em que as máquinas são dispostas na sequência de fabricação de uma família de produtos. É interessante ressaltar que nesse tipo de arranjo físico há a necessidade de mão-de-obra multifuncional, capaz de realizar diversas operações com habilidade [Contador (1995)]. A Figura 1 ilustra um grupo de operadores atuando em uma célula de manufatura.

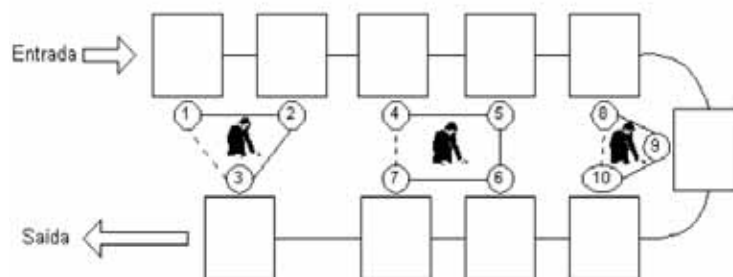


Figura 1 - Arranjo físico celular. Fonte: [Duarte (2003)].

O Problema de Alocação e Roteamento de Operadores em Células de Manufatura (PAROCM) é um problema comumente encontrado em empresas industriais que utilizam células de manufatura como forma de arranjo físico, pois se faz necessário determinar o número de

operadores para atender um determinado volume de produção, alocá-los nas diferentes operações e determinar suas rotas.

Problemas de balanceamento de operações são largamente explorados na literatura, sobretudo Problema de Balanceamento de Linha de Montagem (PBLM) e suas variantes [Klein, (1996)]. Contudo, a literatura sobre problemas que consideram deslocamento entre as operações, caso do PAROCM, é escassa.

O PAROCM consiste em minimizar o número necessário de operadores para realizar diversas operações em uma célula de manufatura, além de determinar as rotas que estes operadores devem seguir. Cada uma das operações deve ser realizada por somente um operador. Entretanto cada operador pode realizar diversas operações, desde que não ultrapasse o *takt-time* (o tempo limite disponível para a produção de um produto de modo que a demanda do cliente seja atendida) da célula em que trabalha.

Além do tempo de realização das tarefas, também é considerado o tempo de deslocamento do operador entre os postos nos quais as diferentes operações são realizadas. O tempo de deslocamento depende da operação de origem e de destino, sendo considerado simétrico. Depois de realizadas todas as operações de um ciclo, cada operador deverá retornar ao seu posto inicial para iniciar o seu próximo ciclo de trabalho.

Algumas variações e restrições adicionais podem ser requeridas neste problema. A mais comum é a restrição de precedência, na qual as operações devem ser realizadas em certa ordem, isto é, algumas operações precedem necessariamente outras. Além desta, também há o caso em que duas operações não podem ser realizadas pelo mesmo operador, ou, de maneira oposta, um operador que realiza uma operação deve realizar outra obrigatoriamente. Este trabalho abordará o caso mais comum do problema, sem estas restrições adicionais.

De modo a entender melhor as particularidades do PAROCM, tome uma célula com a disposição da Figura 2 (a). Como se percebe, há seis operações que devem ser realizadas de forma a cumprir a produção. O objetivo é decidir o número mínimo de operadores necessários e suas rotas, desde que nenhum deles ultrapasse o *takt-time*. É interessante observar que qualquer operador pode partir de qualquer operação para realizar seu ciclo, entende-se como se todas as operações fossem possíveis pontos de partida. Além disso, há o caso em que um operador realiza apenas uma operação, por exemplo, se esta tiver um tempo muito elevado e não for viável o deslocamento até outra operação. Por isso, no pior cenário do problema teríamos, também, seis operadores.

A Figura 2 (b) mostra uma possível solução para o problema. Pode-se observar que três rotas foram formadas o que indica três operadores realizando o conjunto de tarefas. Para o operador que realiza apenas a OP4 não haveria deslocamento, apenas continuaria em seu posto de trabalho para o próximo ciclo, uma conexão foi apresentada, na Figura 2 (b), para este caso, apenas por questão de clareza.

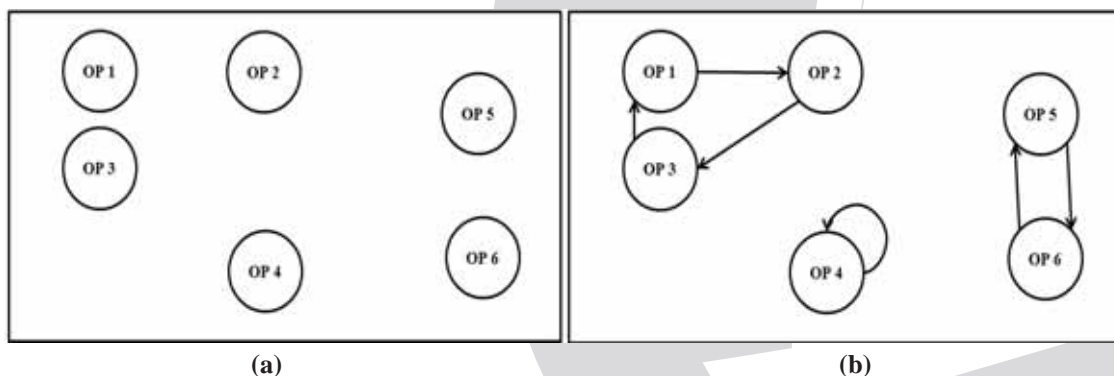


Figura 2 – (a) Exemplo de PAROCM; (b) Possível solução para o PAROCM.

De acordo com o conhecimento dos autores, ainda não há uma formulação matemática na literatura que modele o PAROCM. Por isso, estudaram-se formulações de problemas relacionados para buscar um modelo matemático adequado ao problema tratado. Para um estudo mais completo de problemas relacionados e formulações matemáticas, sugere-se a leitura de [Machado (2014)].

3. Formulação matemática

A formulação apresentada abaixo foi desenvolvida neste trabalho para modelar o PAROCM e segue a seguinte notação:

Parâmetros:

n – número de operações.

D_{ij} – tempo padrão de deslocamento da operação i até a operação j .

T_i – tempo padrão de execução da operação i .

TK – *takt-time* da célula.

Variáveis:

x_{ijk} $\left\{ \begin{array}{l} 1 \text{ Se a operação } j \text{ é realizada imediatamente após a operação } i \text{ pelo operador } k. \\ 0 \text{ Caso contrário.} \end{array} \right.$

Y_i $\left\{ \begin{array}{l} 1 \text{ Se o operador } i \text{ é utilizado.} \\ 0 \text{ Caso contrário.} \end{array} \right.$

Minimizar:

$$\text{Min} \sum_{k=1}^n Y_k \quad (1)$$

Sujeito a :

$$\sum_{i=1}^n \sum_{j=1}^n x_{ijk} \leq n \cdot Y_k \quad \forall K = 1,2,3, \dots, n \quad (2)$$

$$\sum_{k=1}^n \sum_{i=1}^n x_{ijk} = 1 \quad \forall J = 1,2,3, \dots, n \quad (3)$$

$$\sum_{i=1}^n x_{ijk} - \sum_{i=1}^n x_{jik} = 0 \quad \forall J = 1,2,3, \dots, n, K = 1,2,3, \dots, n \quad (4)$$

$$\sum_{i=1}^n \sum_{j=1}^n (x_{ijk} \cdot D_{ij}) + \sum_{i=1}^n \sum_{j=1}^n (x_{ijk} \cdot T_i) \leq TK \quad \forall K = 1,2,3, \dots, n \quad (5)$$

$$U_{ik} - U_{jk} + n \cdot x_{ijk} \leq n - 1 \quad \forall i \neq j \neq K = 1,2,3, \dots, n \quad (6)$$

$$x_{ijk} = (0,1) \quad \forall i, j, k \quad (7)$$

$$Y_k = (0,1) \quad \forall k \quad (8)$$

$$U_{ik} \geq 0 \quad \forall i, k \quad (9)$$

A função objetivo (1) minimiza o número total de operadores. As restrições (2) ativam a variável Y_k se, e somente se, o operador k realiza alguma operação, percebe-se que, como indica a função objetivo, Y_k deve ser minimizado. Restrições (3) estabelecem que deve chegar um, e somente um, operador em cada operação. Restrições (4) são de fluxo e impõem que se um operador tem uma determinada operação como destino, deverá partir desta posteriormente. O conjunto de restrições (5) impede que o tempo total de cada operador, tempo padrão de deslocamento mais tempo padrão de realização das tarefas, ultrapasse o takt-time da célula. As restrições (6) são eliminadoras de subciclo. Por último, restrições (7), (8) e (9) são de integralidade e não negatividade das variáveis do problema.

4. Heurística *multi-start* para o PAROCM

Heurísticas *multi-start* são aquelas que produzem uma solução a cada iteração e a melhor solução é retornada ao final da execução do algoritmo [Martí(2010)]. A heurística proposta neste trabalho, abreviada como HMS, está descrita no Algoritmo 1. O algoritmo constrói rotas utilizando uma Lista de Candidatos (LC) que armazena as operações ainda não visitadas. A primeira operação é selecionada aleatoriamente para compor a primeira rota. Posteriormente, uma Lista de Candidatos Restrita (LCR) é formada com as três operações com menor deslocamento da operação atual. Uma operação é escolhida aleatoriamente da LCR, desde que não ultrapasse o *takt-time* do operador. Quando necessário, uma nova rota é formada. O critério de parada utilizado foi o número de iterações sem melhora.

Algoritmo 1 - Procedimento HMS (itMax)

- 1: **Enquanto** (itSemMelhora < itMax) **faça**
 - 2: Seja $s = \{s^1, \dots, s^n\}$ um conjunto formado por n rotas vazias;
 - 3: Inicialize a LC;
 - 4: Escolha aleatoriamente uma operação k da LC;
 - 5: $s^1 \leftarrow s^1 \cup \{k\}$
 - 6: **Enquanto** (LC $\neq \emptyset$) **faça**
 - 7: Inicialize a LCR;
 - 8: Escolha aleatoriamente uma operação i da LCR;
 - 9: **Se** (Takt-time não for excedido) **faça**
 - 10: $s^v \leftarrow s^v \cup \{i\}$;
 - 11: LC \leftarrow LC $\setminus \{i\}$;
 - 12: **Se não**
 - 13: $v \leftarrow v+1$;
 - 14: **Fim se**
 - 15: **Fim enquanto**
 - 16: **Se** ($f(s) < f(s^*)$) **faça**
 - 17: $s^* \leftarrow s$;
 - 18: **Fim se**
 - 19: **Fim enquanto**
 - 20: Retorne s^* ;
-

5. Resultados

Por não se ter conhecimento de outras abordagens do PAROCM na literatura, não foram encontradas instâncias para *benchmarking* de resolução. Por isso, se fez necessário gerá-las de forma a analisara qualidade da formulação e da heurística proposta. Como visto, será necessário definir o número de operações, o tempo-padrão de realização de cada uma, o tempo-padrão de deslocamento entre as operações e o takt-time da célula.

O PBLM possui instâncias conhecidas e disponíveis para benchmarking e por ser um problema de balanceamento, conta também com número de operações definido e o tempo-padrão de sua execução. Por isso, utilizou-se das instâncias encontradas em [Scholl (2015)] e amplamente utilizadas na literatura do PBLM para a determinação destes parâmetros.

Devido ao elevado número de instâncias disponíveis e a necessidade de se explorar o efeito de outros fatores além do número de operações, foram selecionadas instâncias que se mostravam aparentemente mais coerentes com o objetivo do trabalho, já que algumas instâncias da literatura não parecem ter um número de operações razoável para resolução com formulação matemática, por exemplo, as instâncias nomeadas Barthold e Scholl possuem, respectivamente, 148 e 297 operações. Como lembra [Laporte (2009)] problemas de roteamento de veículos foram só há pouco tempo resolvidos para instâncias superiores a 100 localidades.

Por isso, e para promover certa diversidade de número de tarefas, optou-se por explorar as instâncias presente na Tabela 1. Para o tempo padrão de deslocamento, os dados foram gerados de maneira aleatória no intervalo de 0 a 2 vezes o tempo da operação mais longa, pois em alguns casos operações estão situadas em pontos muito distantes, de modo que, provavelmente, não serão alocadas na mesma rota. O tempo padrão de deslocamento nos casos que o operador só realiza uma tarefa foi definido de acordo com a Equação (10) de forma a garantir que não possa realizar nenhuma outra operação.

$$D_{ii} = TK - T_i \quad (10)$$

Com o objetivo de estudar o impacto do *takt-time*, para cada instância do PBLM escolhida variou-se o *takt-time* em cinco diferentes níveis com base, também, na operação mais longa. Os níveis escolhidos foram: T_{max} , $2 T_{max}$, $4 T_{max}$, $8 T_{max}$, $16 T_{max}$, em que T_{max} é o tempo padrão da operação mais longa. Mais informações sobre as instâncias podem ser encontradas em [Machado (2014)].

Para a resolução do problema através da formulação proposta utilizou-se do software IBM® ILOG® CPLEX® Interactive Optimizer 12.5.0.0 em Ubuntu 12.04 LTS. O computador utilizado possui Intel® Core™ i7-2600K com 8 núcleos, 3.40 GHz por núcleo e 16 GB de RAM. As instâncias foram executadas estabelecendo-se um limite de 10 h de execução. A heurística proposta foi executada dez vezes para cada instância.

A Tabela 1 mostra os resultados computacionais obtidos. Pode-se perceber que em alguns casos o modelo exato apresenta um *gap* não nulo, ou seja, uma variação percentual entre o valor encontrado para função objetivo e o limite inferior fornecido pelo *software*. Um *gap* não nulo indica que não foi possível garantir que a solução encontrada é ou não ótima.

A formulação exata atingiu o valor ótimo em 11 das 20 instâncias e a heurística em 12 das 20 instâncias. Percebe-se que mesmo em alguns casos com *gap* percentual elevado, encontra-se boas soluções no tempo determinado em várias instâncias. Por exemplo, nas instâncias Mitchell com *takt-time* igual a $2 T_{max}$ e $4 T_{max}$ o modelo e a heurística atingiram a diferença de apenas um operador entre a solução encontrada e o limite inferior, não sendo possível determinar se as soluções encontradas são ótimas.

| Instância | N | Takt-time | Limite Inferior | Resolução exata | | | HMS | | | |
|-----------|----|-----------|-----------------|-----------------|--------------------|-----------|-------------|------------|---------------------|-----------------------|
| | | | | Sol. | Gap (%) | Tempo(s) | Melhor sol. | Sol. Média | Gap (%) melhor sol. | Tempo melhor sol. (s) |
| Jackson | 11 | 7 | 10 | 10 | 0,00% [†] | 0,11 | 10 | 10 | 0,00% [†] | 0,03 |
| Jackson | 11 | 14 | 6 | 6 | 0,00% [†] | 0,04 | 6 | 6 | 0,00% [†] | 0,03 |
| Jackson | 11 | 28 | 3 | 3 | 0,00% [†] | 0,77 | 3 | 3 | 0,00% [†] | 0,03 |
| Jackson | 11 | 56 | 2 | 2 | 0,00% [†] | 0,46 | 2 | 2 | 0,00% [†] | 0,02 |
| Jackson | 11 | 112 | 1 | 1 | 0,00% [†] | 0,19 | 1 | 1 | 0,00% [†] | 0,02 |
| Mitchell | 21 | 13 | 14 | 14 | 0,00% [†] | 0,37 | 14 | 14 | 0,00% [†] | 0,06 |
| Mitchell | 21 | 26 | 7 | 8 | 12,50% | 36000 | 8 | 8,7 | 12,50% | 0,05 |
| Mitchell | 21 | 52 | 3 | 4 | 25,00% | 12007,19* | 4 | 4 | 25,00% | 0,05 |
| Mitchell | 21 | 104 | 2 | 2 | 0,00% [†] | 83,67 | 2 | 2 | 0,00% [†] | 0,04 |
| Mitchell | 21 | 208 | 1 | 1 | 0,00% [†] | 1,39 | 1 | 1 | 0,00% [†] | 0,04 |
| Sawyer | 30 | 25 | 22 | 22 | 0,00% [†] | 0,93 | 22 | 22 | 0,00% [†] | 0,09 |
| Sawyer | 30 | 50 | 9 | 11 | 18,18% | 36000 | 12 | 12,8 | 25,00% | 0,09 |
| Sawyer | 30 | 100 | 5 | 7 | 28,57% | 36000 | 6 | 6,8 | 16,67% | 0,07 |
| Sawyer | 30 | 200 | 3 | 4 | 25,00% | 36000 | 3 | 3 | 0,00% [†] | 0,06 |
| Sawyer | 30 | 400 | 2 | 2 | 0,00% [†] | 204,24 | 2 | 2 | 0,00% [†] | 0,06 |
| Warnecke | 58 | 53 | 41 | 41 | 0,00% [†] | 379,91 | 42 | 42,9 | 2,38% | 0,27 |
| Warnecke | 58 | 106 | 17 | 22 | 22,73% | 36000 | 25 | 26,1 | 32,00% | 0,26 |
| Warnecke | 58 | 212 | 9 | 22 | 59,09% | 36000 | 12 | 13 | 25,00% | 0,23 |
| Warnecke | 58 | 424 | 5 | 11 | 54,55% | 36000 | 6 | 6 | 16,67% | 0,20 |
| Warnecke | 58 | 848 | 3 | 7 | 57,14% | 36000 | 3 | 3 | 0,00% [†] | 0,19 |

* Processamento fechado pelo sistema operacional devido ao excesso de consumo de RAM. † Resultado ótimo.

Tabela 1 – Resultados computacionais

Pode-se perceber que há grande influência do *takt-time* na dificuldade imposta pela instância. A Figura 3 mostra uma análise desta influência para a instância Mitchell na resolução exata do problema. Há claramente uma faixa crítica do parâmetro entre 26 e 58 na qual não foi possível garantir a otimalidade da solução encontrada. Isso ocorre, provavelmente, pois valores intermediários de *takt-time* aumentam o número de combinações viáveis. Por outro lado, valores muito elevados tornam o problema trivial, pois fica elementar criar uma rota com poucos operadores.

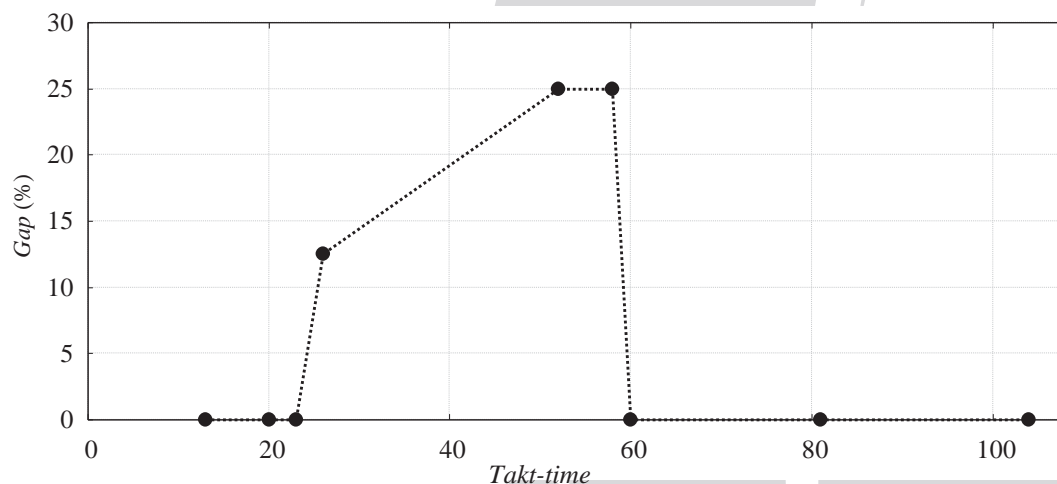


Figura 3 – Análise da influência do *takt-time* para instância Mitchell.

6. Conclusões e trabalhos futuros

Neste trabalho foi introduzido o PAROCM, problema de elevado interesse no ambiente industrial e, que de acordo com o conhecimento dos autores, não havia sido abordado por meio de técnicas de otimização. Por outro lado, no ambiente acadêmico, o presente estudo preenche uma lacuna na modelagem e resolução de problemas de balanceamento de operadores em que o deslocamento entre as operações precisa ser computado.

As técnicas de resolução propostas auxiliam os analistas das indústrias em uma tarefa extremamente complexa que dificilmente encontrariam a solução ótima de forma manual. Contudo, a experiência e a sensibilidade dos analistas podem ser usadas para aprimorar o resultado obtido com a aplicação do modelo, por exemplo, promovendo a redução dos deslocamentos entre operações.

Neste contexto, a principal contribuição deste artigo é a formulação matemática para o problema abordado. Adicionalmente foi proposta uma heurística simples, para que através de simulações, possamos ter uma alternativa de solução menos custosa, e apropriada para elevadas dimensões. Outra contribuição deste trabalho foi a geração de um conjunto de instâncias para o PAROCM e uma análise preliminar dos parâmetros que tende a indicar quando uma instância é mais difícil (Figura 3).

Os resultados computacionais obtidos mostraram que é possível, para algumas instâncias, a resolução exata em tempos de execução aceitáveis. Isto foi realizado, inclusive, para uma instância relativamente grande, com 58 operações. Contudo, o modelo mostrou sensibilidade ao parâmetro *takt-time* impossibilitando a resolução em alguns casos, até para instâncias menores. Propõe-se para trabalhos futuros, a geração de desigualdades válidas para melhorar o desempenho da abordagem exata deste trabalho. Além disso, espera-se que melhores limites inferiores sejam obtidos para verificar a otimalidade de soluções encontradas neste trabalho.

Agradecimentos

Os autores gostariam de agradecer o apoio da CAPES, CNPq e da FAPERJ.

Referências

- Contador, J. C.** (1995), Células de manufatura, *Revista Produção*, 5, 45-64.
- Duarte, R. N.** Simulação Computacional: Análise de uma Célula de Manufatura em Lotes do Setor de Auto-Peças, *Dissertação de mestrado - Universidade Federal de Itajubá*, 2003.
- Klein, R. e Scholl, A.** (1996), Maximizing the production rate in simple assembly line balancing-A branch and bound procedure, *European Journal of Operational Research*, 91, 367-385.
- Laporte, G.** (2009), Fifty years of vehicle routing. *Transportation Science*, 43, 408-416.
- Machado, V.** Formulação e resolução exata do problema de alocação e roteamento de operadores em células de manufatura, *Trabalho de Conclusão de Curso - Escola de Engenharia Industrial Metalúrgica de Volta Redonda, Universidade Federal Fluminense, Volta Redonda, Brasil, 2014.*
- Martí, R., Moreno-Vega, J., Duarte, A.** Advanced multi-start methods. In: *Handbook of metaheuristics*. Springer US, 2010.
- Nagy, G. e Salhi, S.** (2007), Location-routing: Issues, models and methods, *European Journal of Operational Research*, 177, 649-672.
- Scholl, A.** (2015), Assembly line optimization research. Disponível em: <http://www.assembly-line-balancing.de>. Acesso em: 10 fev. 2015.