



Resumo Expandido

Título da Pesquisa: Inteligência Computacional para a Resolução de Problemas Combinatórios		
Palavras-chave: Pesquisa Operacional; Heurísticas; Otimização Combinatória.		
Campus: Governador Valadares	Tipo de Bolsa: PIBIC	Financiador: IFMG
Bolsista (as): Lauren Isis Cunha e Victor Correa Viana		
Professor Orientador: Guido Pantuza Júnior		
Área de Conhecimento: Engenharia de Produção – Pesquisa Operacional		

Resumo: A rotina de um Engenheiro de Produção é caracterizada pela resolução de problemas que envolvem a tarefa de alocar os fatores de produção, tais como: programação de horários nas escolas; alocação de salas de aula; seleção de qual cultura deve ser cultivada; sequenciamento da produção; localização de instalações industriais. Esses problemas possuem variáveis de decisão inteiras, por isso eles são conhecidos como problemas discretos ou combinatórios. O objetivo deste projeto de pesquisa envolve algoritmos heurísticos de otimização para a resolução destes problemas combinatórios, com grande potencial de aplicação a problemas reais. Propõe-se desenvolver o estudo, formulação, implementação, avaliação, e análise desses algoritmos aplicados aos problemas combinatórios e sua comparação com modelos de programação matemática. O método heurístico proposto foi baseado na metaheurística VNS e seus resultados foram comparados com um modelo de programação matemática. Observando os resultados, nota-se que o VNS teve um excelente desempenho.

INTRODUÇÃO:

A rotina de um Engenheiro de Produção é caracterizada pela resolução de problemas que envolvem a tarefa de alocar os fatores de produção, tais como mão-de-obra, terra, capital e matéria-prima. Esta alocação deve ser realizada de forma otimizada para garantir a eficiência do processo produtivo.

Entre os diversos fatores de produção, um dos que possuem maior impacto nos custos produtivos é a mão-de-obra. Segundo a FIESP (2011), a mão de obra, em 2009, foi responsável, em média, por 32,4% do custo total dos bens produzidos no Brasil. Trata-se do valor mais alto de todos os países estudados (34 maiores economias) e 11% acima da média mundial que foi de 21,4%. Se comparado com outros países em desenvolvimento temos: 14,7% em Taiwan, 17% na Argentina e Coreia do Sul e 27% no México. Isto se deve ao fato dos recentes aumentos dos salários acima da inflação, adicionados aos encargos sociais e benefícios. Por isso, as empresas procuram a redução do número de funcionários sem afetar os prazos de entrega e qualidade do produto final.

Esse problema enfrentado pelas empresas é conhecido na literatura como o problema de sequenciamento com alocação de trabalhadores (*Scheduling Problem with Worker Allocation – SPWA*). O problema consiste em alocar as tarefas aos trabalhadores de uma empresa, minimizando o instante de término da última tarefa executada (*makespan*) e o número de funcionários utilizados.

O problema é composto por um conjunto de trabalhadores, *Worker*, e um conjunto de tarefas, *Job*. Ele consiste em alocar as tarefas aos trabalhadores e propor uma sequência de execução, respeitando as restrições de qualificação. Ou seja, um trabalhador só pode executar uma tarefa se ele for qualificado. Além

disso, cada trabalhador, para executar uma mesma tarefa, necessita de tempos diferentes de acordo com suas habilidades.

Neste trabalho consideramos as seguintes restrições: Todas as tarefas devem ser executadas. Não consideramos o tempo de deslocamento dos funcionários ou o tempo de espera em fila. O horizonte de planejamento da alocação dos trabalhadores é fixo. Cada tarefa é executada por apenas um único funcionário. O excesso de trabalho para os trabalhadores não é considerado, ou seja, não consideramos que os trabalhadores ficarão sobrecarregados com excesso de trabalho. Todas as tarefas tem o mesmo nível de esforço. Todos os funcionários possuem o mesmo custo (salário). Cada colaborador possui uma habilidade diferente, ou seja, cada um executa a mesma tarefa com um tempo diferente.

Uma das ferramentas usadas para a otimização de recursos é a Pesquisa Operacional (PO). Ela teve sua origem durante a Segunda Guerra Mundial e também é chamada de ciência e tecnologia de decisão. Ela consiste, segundo Arenales *et al.* (2007), na aplicação de métodos científicos para auxiliar no processo de tomada de decisões, tais como e operar sistemas que requerem alocações eficientes de recursos escassos.

Uma das técnicas de PO mais utilizadas para a resolução de problemas combinatórios, assim como o SPWA, é a programação matemática. Uma outra técnica de PO muito utilizada dentro deste contexto é a utilização de inteligência computacional através de métodos heurísticos.

Na programação matemática, obtém-se uma única solução ótima segundo algum critério pré-estabelecido. Já no método heurístico, busca-se uma solução eficiente próxima da solução ótima global. O uso dos métodos heurísticos é recomendado quando não há tempo hábil para a resolução do problema através do modelo de programação matemática.

Na literatura encontramos alguns trabalhos que abordam este problema. Lima e Sannomiya (2001), propuseram uma heurística para resolução do SPWA fundamentada no Algoritmo Genético chamada *Module Type Genetic Algorithm* (MTGA).

Lima e Sannomiya (2002), também utilizaram o MTGA, porém consideraram que cada trabalhador possui o mesmo nível de qualificação para executar as diferentes tarefas.

Osawa e Ida (2005), também utilizaram Algoritmo Genético, porém propuseram um novo método de seleção da população sobrevivente.

METODOLOGIA:

Para a resolução do SPWA propõe-se o modelo de programação matemática a seguir:

Sendo os parâmetros de entrada:

T_{ij} : Tempo necessário para o funcionário i executar a tarefa j

E as variáveis de decisão:

$$x_{ij} : \begin{cases} 1 \text{ se o funcionário } i \text{ executa a tarefa } j \\ 0 \text{ caso contrário} \end{cases}$$
$$y_i : \begin{cases} 1 \text{ se o funcionário } i \text{ é utilizado} \\ 0 \text{ caso contrário} \end{cases}$$

A Função Objetivo é apresentada pela Eq. (1):

$$\min f = \left(\sigma + \sum_{i \in \text{Pessoas}}^n y_i \right) \quad (1)$$

A Eq.(2) garante que toda tarefa deve ser executada apenas uma vez.

$$\sum_{i \in \text{Pessoas}} x_{ij} = 1 \quad \forall j \in \text{Tarefas} \quad (2)$$

A Eq.(3) diz respeito ao *makespan*.

$$\sum_{j \in \text{Tarefas}} T_{ij} x_{ij} \leq \sigma \quad \forall i \in \text{Pessoas} \quad (3)$$

A Eq.(4) diz respeito ao número de pessoas utilizadas.

$$\frac{\sum_{j \in \text{Tarefas}} x_{ij}}{|\text{Tarefas}|} \leq y_i \quad \forall i \in \text{Pessoas} \quad (4)$$

As Eq.(5) a Eq.(7) garantem o domínio das variáveis.

$$x_{ij} \in \{0,1\} \quad \forall i \in \text{Pessoas e } j \in \text{Tarefas} \quad (5)$$

$$y_i \in \{0,1\} \quad \forall i \in \text{Pessoas} \quad (6)$$

$$\sigma \in \mathbb{R} \quad (7)$$

Além do modelo de programação matemática anterior, também utilizamos o método heurístico VNS. O método heurístico VNS, exemplificado pela Figura 1, proposto por Mladenovic e Hansen (1997), é um método que consiste em explorar o espaço de soluções por meio de trocas sistemáticas das estruturas de vizinhança. Contrariamente às outras metaheurísticas baseadas em métodos de busca local, o método VNS não segue uma trajetória. Ele explora vizinhanças diferentes da solução corrente e focaliza a busca em torno de uma nova solução se, e somente se, um movimento de melhora é realizado.

O VNS também inclui um procedimento de busca local a ser aplicado sobre a solução corrente. No presente trabalho adotou-se o método VND, exemplificado pela Figura 2 para fazer a busca local, também utilizado pelo VNS original.

```

Procedimento VNS ( $f(\cdot), N(\cdot), r, s$ );
1  Seja  $s_0$  uma solução inicial;
2  Seja  $r$  o número de estruturas diferentes de vizinhança;
3   $s \leftarrow s_0$ ;
4  Enquanto ( critério de parada não satisfeito ) faça
5       $k \leftarrow 1$ ;
6      Enquanto (  $k \leq r$  ) faça
7          Gere um vizinho qualquer  $s' \in N^{(k)}(s)$ ;
8           $s'' \leftarrow \text{BuscaLocal}(s')$ ;
9          se (  $f(s'') < f(s)$  ) então
10              $s \leftarrow s''$ ;
11              $k \leftarrow 1$ ;
12          Senão
13              $k \leftarrow k + 1$ ;
14          Fim se
15          Fim-enquanto;

```

```

16 Fim-enquanto;
17 Retorne s;
Fim VNS;

```

Figura 1: Heurística VNS

```

Procedimento VND ( $f(\cdot), N(\cdot), r, s$ );
1  Seja  $r$  o número de estruturas diferentes de vizinhança;
2   $k \leftarrow 1$ ;
3  Enquanto ( $k \leq r$ ) faça
4      Encontre o melhor vizinho  $s' \in N^{(k)}(s)$ ;
5      se ( $f(s') < f(s)$ ) então
6           $s \leftarrow s'$ ;
7           $k \leftarrow 1$ ;
8      Senão
9           $k \leftarrow k + 1$ ;
10     Fim se
11 Fim-enquanto;
12 Retorne  $s$ ;
Fim VND;

```

Figura 2: Heurística VND

Representação de uma solução

Uma solução pode ser representada por uma matriz S , exemplificada pela Figura 3. A primeira coluna apresenta os trabalhadores. A coluna “**Util**” indica se o funcionário i realiza uma tarefa j qualquer. As colunas “**Tarefas**” indicam as tarefas alocadas a cada trabalhador e sua respectiva sequência.

		Util	Tarefas				
Trabalhador	1	1	1	3	4	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	1	2	5	0	0	0

Figura 3: Representação de uma solução S .

Geração da solução Inicial

A solução inicial gera $NSol$ matrizes s_i através de um procedimento parcialmente guloso utilizando um torneio binário.

Para cada matriz solução S , a cada iteração do procedimento, uma tarefa j é selecionada. Além disso, dois funcionários também são escolhidos aleatoriamente. Aquele funcionário que executar a tarefa j selecionada em menor tempo é escolhido. Este procedimento é executado até que todas as tarefas sejam atribuídas a algum funcionário.

Avaliação de uma solução

Uma solução S é avaliada segundo a função de avaliação $f(S)$. Esta função busca minimizar o número de funcionários e *makespan* através de penalidades. A função de avaliação $f(S)$ é exemplificada pela Eq. (8).

$$\min f(s) = \frac{w_l}{\Delta^\sigma} \sigma(s) + \left(\frac{1-w_l}{\Delta^\tau} \right) \tau(s) \quad (8)$$

Sendo:

- w_l : Penalidade da solução s_l .
- σ : Instante de término da última tarefa executada.
- τ : Número de trabalhadores utilizado.
- Δ^σ : Fator de correção para o parâmetro σ .
- Δ^τ : Fator de correção para o parâmetro τ .

Estruturas de vizinhança

Para explorar o espaço de soluções do problema, ou seja, para encontrar uma solução S' , dita vizinha de S , foram utilizados dois movimentos apresentados a seguir:

Movimento Realoca Tarefa, $N^{RT}(s_l)$: Este movimento consiste em escolher aleatoriamente um trabalhador i que está sendo utilizado. Em seguida, a última tarefa j é selecionada e realocada para outro trabalhador escolhido aleatoriamente.

Movimento Trabalhador, $N^T(s_l)$: Este movimento consiste em escolher aleatoriamente um trabalhador i que está sendo utilizado e realocar todas as suas tarefas para os demais trabalhadores, aleatoriamente.

RESULTADOS E DISCUSSÕES:

Para testar os modelos propostos foram utilizadas 4 instâncias-teste. Elas podem ser encontradas em http://www.4shared.com/zip/m7xmUfpi/Instance_-_SPWA.html. Elas diferem entre si pelo número de trabalhadores e de tarefas. A tabela 1 apresenta algumas características das instâncias. As colunas **#Worker** e **#Job** mostram, respectivamente, o número de trabalhadores e o número de tarefas que precisam ser executadas.

O algoritmo VNS e o modelo proposto foi desenvolvido na linguagem C, usando o compilador C++ *Builder 5.0* da *Borland*. Ambos os modelos foram testados em um PC *Pentium Core 2 Duo*, com 2,4 GHz e 4 GB de RAM sob plataforma *Windows 7*. A tabela 1 apresenta algumas características das instâncias. As colunas **#Worker** e **#Job** mostram, respectivamente, o número de trabalhadores e o número de tarefas que precisam ser executadas.

Tabela 1: Características das instâncias-teste

	#Worker	#Job
Instância 1	5	10
Instância 2	10	50
Instância 3	15	100
Instância 4	25	100

A tabela 2 apresenta os resultados encontrados. Na linha *Exato*, temos os resultados para os valores do modelo de programação matemática, e na linha *VNS*, os resultados médios do modelo heurístico, encontrados após 100 execuções. Na coluna *#Trab* tem-se o número de trabalhadores, na coluna *Makespan*, o tempo total gasto para executar todas as tarefas (em minutos). Na coluna *Tempo*, tem-se o tempo gasto pelo modelo para encontrar o resultado, em segundos. Observa-se que os resultados do VNS proposto estão próximos do modelo de programação matemática. Além disso, ele só não foi mais rápido na *Instância 2*.

Tabela 2: Resultados.

Instância 1			Instância 2			
	# Trab	Makespan	Tempo (seg)	# Trab	Makespan	Tempo (seg)
<i>Exato</i>	5	8	2	10	20	8
<i>VNS</i>	4	9	1,6	8	28	25,4
Instância 3			Instância 4			
	# Trab	Makespan	Tempo (seg)	# Trab	Makespan	Tempo (seg)
<i>Exato</i>	15	27	183	20	20	47.696,0
<i>VNS</i>	13	41	110,2	14	32	164,8

CONCLUSÕES:

Pelos testes realizados percebe-se que o algoritmo proposto é capaz de encontrar uma boa solução para o problema em estudo. O algoritmo demonstrou-se eficiente gerando boas soluções, com valores próximos na quase totalidade dos testes. Uma proposta para trabalhos futuros é a utilização de novos movimentos para o SPWA.

REFERÊNCIA BIBLIOGRÁFICA:

Arenales, M. et al. (2007). *Pesquisa operacional: para cursos de engenharia*. Rio de Janeiro: Elsevier.

FIESP (2011). Encargos trabalhistas sobre folha de salários e seus impactos no Brasil e no mundo. Relatório Técnico. Fiesp/Decomtec. São Paulo – SP, 2011.

Lima, H. e Sannomiya, N. (2001). Module Type Genetic Algorithm for Modified Scheduling Problems with Worker Allocation. *Anais do American Control Conference*. Arlington, VA.

Lima, H. e Sannomiya, N. (2002). Proposition of Module Type Genetic Algorithm and Its Application to Modified Scheduling Problems with Worker Allocation. *IEEE Japan*, v.122-C, p. 409-416.

Mladenovic, N. e Hansen, P. (1997). Variable Neighborhood Search. *Computers and Operations Research*, n. 24 pp. 1097–1100.

Osawa, A. e Ida, K. (2005). Scheduling Problem with Worker Allocation using Genetic Algorithm. *Japan-Australia workshop on intelligent and evolutionary systems*, pp.1-8.

PARTICIPAÇÃO EM CONGRESSOS, PUBLICAÇÕES E/OU PEDIDOS DE PROTEÇÃO INTELECTUAL:

Trabalhos aceitos para apresentação em congressos nacionais:

Pantuza Jr, G. (2012). Um modelo de programação matemática multiobjetivo para o problema de sequenciamento e alocação de trabalhadores. Simpósio de Pesquisa Operacional e logística da Marinha 2012, Rio de Janeiro, RJ.

Pantuza Jr, G. (2012). Um algoritmo VNS multiobjetivo para o problema de sequenciamento com alocação de trabalhadores. Simpósio Brasileiro de Pesquisa Operacional – SBPO 2012. Rio de Janeiro, RJ.

Trabalhos aceitos para apresentação em congressos internacionais:

Pantuza Jr, G. (2012). Um algoritmo VNS multiobjetivo para o problema de sequenciamento com alocação de trabalhadores. Congreso Latino-Iberoamericano de Investigación Operativa – CLAIO 2012. Rio de Janeiro, RJ.