

6º CONGRESSO BRASILEIRO DE PESQUISA E DESENVOLVIMENTO EM PETRÓLEO E GÁS



TÍTULO DO TRABALHO:

METAHEURÍSTICA GRASP COM PATH RELINKING APLICADA AO PROBLEMA DE PROGRAMAÇÃO DE EMBARCAÇÕES PLV

AUTORES:

Maciel Manoel Queiroz¹ e André Bergsten Mendes²

INSTITUIÇÃO:

Universidade de São Paulo – EPUSP – Departamento de Engenharia Naval e Oceânica
Av. Prof. Mello Moraes, 2231- Cidade Universitária, 05508-030 – São Paulo, SP

¹maciel.queiroz@usp.br; ²andbergs@usp.br

Este Trabalho foi preparado para apresentação no 6º Congresso Brasileiro de Pesquisa e Desenvolvimento em Petróleo e Gás- 6º PDPETRO, realizado pela Associação Brasileira de P&D em Petróleo e Gás-ABPG, no período de 09 a 13 de outubro de 2011, em Florianópolis-SC. Esse Trabalho foi selecionado pelo Comitê Científico do evento para apresentação, seguindo as informações contidas no documento submetido pelo(s) autor(es). O conteúdo do Trabalho, como apresentado, não foi revisado pela ABPG. Os organizadores não irão traduzir ou corrigir os textos recebidos. O material conforme, apresentado, não necessariamente reflete as opiniões da Associação Brasileira de P&D em Petróleo e Gás. O(s) autor(es) tem conhecimento e aprovação de que este Trabalho seja publicado nos Anais do 6ºPDPETRO.

METAHEURÍSTICA GRASP COM PATH RELINKING APLICADA AO PROBLEMA DE PROGRAMAÇÃO DE EMBARCAÇÕES PLV

Abstract

This research addresses a fleet scheduling problem present in the offshore oil industry. Among the special purpose services one will find the pipe layer activities accomplished by the Pipe Layer Vessels (PLV). The jobs are characterized by a release date, which reflects the expected arrival date of the necessary material at the port. There are compatibility constraints between job and vessel, so that some vessels may not be able to perform a certain job, the duration of the jobs can be differentiated by vessel and if a job is finished after its due date, a penalty is incurred. This is a variation of the unrelated parallel machine problem with weighted tardiness objective function. In this paper it will be addressed a metaheuristic GRASP with Path Relinking, which have proved to be competitive and an effective solution strategy. Computational experiments were conducted, comparing solutions with bounds provided by linear column generation.

KEYWORDS. Scheduling Unrelated Parallel Machine, Grasp, Path Relinking.

1. Introdução

A área de programação, conhecida na literatura técnica como “scheduling”, está presente em diversos sistemas produtivos, e sempre envolve um número de atividades que estão atreladas à utilização de recursos por um determinado período de tempo (Morton e Pentico, 1993). A programação de tarefas e recursos possui um importante papel na maioria dos sistemas de manufatura, produção e ambientes de processamento de informação, como também nas áreas de transportes e em outros diversos tipos de indústrias (Pinedo, 2002).

A complexidade do processo de programação de tarefas e recursos é grande tendo em vista a natureza combinatória destes problemas, associada às diversas restrições que devem ser observadas, além da escala dos problemas reais. Este trabalho irá focar o problema de programação de uma frota de navios especializados no processo de lançamento de “risers” e interligação submarina, com o uso de técnicas heurísticas.

2. Descrição do Problema

A exploração e produção de petróleo em alto mar, conhecida como exploração “offshore”, necessita de uma grande quantidade de atividades de apoio marítimo, requerendo a utilização dos mais variados tipos de embarcações especializadas. Uma etapa importante do desenvolvimento de um campo petrolífero é a de lançamento de “risers” ou linhas de produção. As embarcações responsáveis pelo lançamento dos dutos e sua interligação são as PLVs (“pipe layer vessels”), que são recursos críticos que uma empresa petrolífera tem que administrar.

Para a instalação de uma linha, seja ela flexível ou rígida, existem diversas atividades necessárias para que isto se concretize. No início da operação a embarcação necessita fazer o carregamento das linhas e de equipamentos que serão utilizados na operação de lançamento. O próximo passo consiste em deslocar-se até ao local de instalação, dando início ao lançamento. Dependendo da profundidade e da extensão do lançamento, pode ser necessário retornar novamente ao porto para carregar material adicional. Existem casos de maior complexidade, em que é necessária a requisição de embarcações de apoio como, por exemplo, um rebocador (AHTS) ou uma embarcação dotada de sonda de observação (RSV).

Quando uma empresa petrolífera realiza o planejamento operacional da sua frota de PLVs, considera como uma macro-tarefa todas as atividades acima mencionadas. Em geral, o setor responsável pelo planejamento e execução desta operação avalia quais embarcações da frota estão

aptas para realizar cada macro-tarefa (doravante, denominado simplesmente de tarefa), e elaboram uma matriz de compatibilidade entre cada par embarcação-tarefa, estimando o tempo necessário para que a tarefa seja realizada.

Postergar uma tarefa, para realizar outra, implica em adiar o retorno sobre o investimento realizado. Para programar a frota, algumas restrições operacionais devem ser observadas, como a compatibilidade embarcação-tarefa e a data mais cedo em que a tarefa pode ser iniciada. A extensão do horizonte de planejamento também deve ser considerada, já que a empresa trabalha com uma visão de médio prazo, que no caso da operação de lançamento de “risers” e interligação constitui um período de 3 a 6 meses. É importante ressaltar, contudo, que este horizonte, em princípio, não é conhecido com precisão. Os técnicos responsáveis pelo setor estimam um prazo capaz de atender a demanda e usa este horizonte como meta.

Objetiva-se achar a seqüência de execução das tarefas, de forma a minimizar as perdas financeiras pelo atraso no início da execução das mesmas em relação aos respectivos instantes de liberação. Se o horizonte de planejamento for violado, existe uma grande penalidade associada a este tipo de caso. Quanto à caracterização da demanda, adota-se como hipótese que a mesma é conhecida, bem como a duração da execução das tarefas. No caso da oferta, os recursos empregados são as embarcações, que se diferenciam uma das outras geralmente pela capacidade de armazenagem e tipo de lançamento das linhas. Admite-se conhecida a composição da frota, os instantes e os locais em que as embarcações estão disponíveis.

Este problema tem semelhanças com o problema de programação de máquinas paralelas não-relacionadas, com minimização do atraso ponderado e instante de liberação das tarefas. Dado que algumas embarcações não estão habilitadas a realizar determinados serviços e, pelo fato das embarcações poderem realizar as tarefas em tempos diferentes, fica caracterizado o aspecto das máquinas serem não-relacionadas.

3. Revisão da Literatura

Devido o problema deste trabalho ser uma variação da classe de problemas de máquinas paralelas não-relacionadas, podemos destacar algumas contribuições de trabalhos nos últimos anos. Suponha que m máquinas M_i ($i = 1, \dots, m$) tenham que processar n tarefas J_j ($j = 1, \dots, n$). A programação é responsável pela alocação em um ou mais intervalo de tempo para uma ou mais máquinas (Brucker, 2007).

Fuller (1978) faz uma comparação entre soluções ótimas e soluções boas, analisando as heurísticas no processo de decisão. O autor destaca que as heurísticas simplificam o processo para o tomador de decisão, permitindo que as decisões sejam tomadas de forma rápida. Alguns autores ao longo dos anos têm investido tempo em problemas de programação de máquinas paralelas, como por exemplo, em casos de função objetivo de atraso total ponderado: Liaw et al. (2003), Alidaee e Rosa (1997); Atraso total: Shim e Kim (2007), Biskup et al. (2008) e Koulamas (1997). Para maiores detalhamentos sobre as variações de problemas de máquinas paralelas, Cheng e Sin (1990) fazem uma revisão do estado-da-arte em pesquisas de máquinas paralelas.

3.1. Metaheurística Grasp

Segundo Feo e Resende (1989, 1995), a meta-heurística GRASP (Greedy Randomized Adaptive Search Procedure) trabalha com múltiplas soluções iniciais e, em cada iteração, duas fases devem ser realizadas: fase de construção e busca local. Na primeira fase é construída uma solução viável, por meio de um algoritmo guloso aleatório; na segunda fase, a vizinhança da solução gerada é explorada, até que um ótimo local seja atingido.

Inicialmente, é requerido uma lista ordenada de tarefas de acordo com algum critério. Em seguida, constrói-se uma lista restrita de candidatos (RCL), que tem como objetivo armazenar os melhores elementos que possam fazer parte da solução parcialmente gerada, a qual será usada para sorteio da

“próxima” tarefa que será incorporada à solução (parcial). Uma vez sorteado, a lista é refeita e o processo continua até que todas as tarefas tenham sido sorteadas.

3.2. Path Relinking

A heurística Path Relinking ou (Reconexão por Caminhos), descrita originalmente por Glover (1996), e também em Glover et al. (2000), consiste em explorar o caminho ou trajetória entre duas soluções, uma denominada de solução inicial e a outra de solução guia. Em cada iteração é realizado um movimento na solução inicial, com o objetivo de aproximá-la da solução guia. Isto implica em introduzir atributos encontrados na solução guia na solução inicial, até que ambas as soluções sejam iguais. Esta trajetória pode ser compreendida como um processo de intensificação e é importante por permitir explorar espaços de solução neste caminho de aproximação entre as duas soluções. É possível que, ao longo deste processo, um novo ótimo local seja alcançado, conforme ilustra a figura 1, onde a solução inicial é dada por x' e a solução guia x'' . Para chegar de x' a x'' são produzidas novas soluções: $x' = x(1), x(2), \dots, x(r) = x''$, indicada pela linha pontilhada.

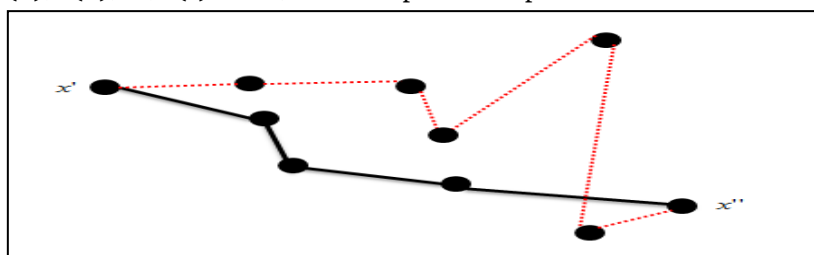


Figura 1. Exemplo Path Relinking

Resende e Ribeiro (2003) destacam algumas variações importantes do Path Relinking: Forward Relinking: a trajetória é construída partindo da solução inicial x' , até alcançar a solução guia x'' ; Backward Relinking: esta estratégia é o inverso da anterior, ou seja, a solução inicial neste caso é x'' e a guia x' ; Mixed Relinking: duas trajetórias são simultaneamente exploradas, a primeira iniciando em x' e a segunda em x'' , até que se encontrem.

Em Resende et. al. (2010) é discutido o Randomized Path Relinking (RPR). A versão usual do Path Relinking adota um critério guloso na escolha do melhor movimento em cada iteração, na construção da trajetória que ligará a solução inicial à guia. Isto faz com que, certamente, um único caminho será gerado, o que poderá impedir a obtenção de soluções melhores. O RPR introduz uma lista restrita de candidatos, na qual os melhores movimentos são inseridos e um deles, aleatoriamente, é escolhido. Isto garante que trajetórias diferentes serão geradas, aumentando a possibilidade de atingir novas soluções, eventualmente, de boa qualidade.

4. Metodologia e Modelagem Matemática

Para o presente problema, foi formulado um modelo de programação linear inteira mista, usando as idéias apresentadas em Van Den Akker, et al. (2000), para um problema de máquina simples com função objetivo regular. Esta formulação baseia-se em uma escala de tempo discretizada. É assumido que os parâmetros são conhecidos e inteiros, sendo utilizada a notação de Unlu e Masson (2010).

Conjuntos

J	Conjunto de tarefas	$j = 1, \dots, n$
M	Conjunto de embarcações	$i = 1, \dots, m$

Parâmetros

	Horizonte de planejamento = T (índice t)
r_j	Instante de liberação da tarefa j

- p_j^i Tempo de execução da tarefa j pela embarcação i
 c_{jt} Penalidade associada a tarefa j , quando é iniciada no instante t
 a_{ij} Parâmetro binário que será igual a 1 se a tarefa j for compatível com a embarcação i

Variáveis de Decisão

$x_{ij}^t = 1$ Se a tarefa $j \in J$ é designada para a embarcação $i \in M$ no instante t ; 0 em caso contrário.

Modelo

$$\min \sum_{i=1}^m \sum_{j=1}^n \sum_{t=r_j}^{T-p_j^i+1} c_{jt} x_{ij}^t \quad (1)$$

Sujeito a:

$$\sum_{i=1}^m \sum_{t=r_j}^{T-p_j^i+1} a_{ij} x_{ij}^t = 1 \quad \forall j \quad (2)$$

$$\sum_{j=1}^n \sum_{s=t-p_j^i+1}^{T-p_j^i+1} x_{ij}^s \leq 1 \quad \forall t, i \quad (3)$$

$$x_{ij}^t \in \{0,1\} \quad \forall j, t, i \quad (4)$$

A equação (1) é a função objetivo do problema e contempla a soma dos atrasos ponderados de todas as tarefas. A restrição (2) garante que toda a demanda seja atendida, por meio de embarcações compatíveis. A restrição (3) garante que, em cada instante, não seja violada a capacidade da embarcação. A equação (4) impõe que as variáveis sejam binárias.

Este modelo foi utilizado para gerar limitantes inferiores para os cenários de teste, por meio da técnica de geração de colunas. Utilizou-se a estrutura apresentada em Van Den Akker, et al. (2000), em que o problema foi reformulado utilizando-se a decomposição de Dantzig-Wolfe.

5. Heurísticas Propostas

Para o presente trabalho foi realizada uma implementação da heurística GRASP e, posteriormente, adicionou-se o mecanismo de Path Relinking. A versão inicial segue a estrutura apresentada a seguir.

```

Procedimento GRASP (maxIterações)
1   LerDados ()
2   Para  $k=1, \dots, \text{maxIterações}$  faça
3       Atualiza configuração  $k$ 
4       Para  $j=1, \dots, n$  faça
5           Atualiza lista restrita de candidatos de tamanho  $l(k)$ 
6           Sorteia uma tarefa  $i$  em  $RCL(k)$ 
7           Inserir a tarefa  $i$  na melhor posição viável de inserção
8       Fim
9       BuscaLocal ()
10      Atualiza MelhorSolução
11  Fim
12  Retorna MelhorSolução
Fim GRASP

```

Figura 2 – Estrutura básica do GRASP

Em cada iteração será atualizada em uma determinada “configuração” do algoritmo. Esta configuração consiste em três elementos: o tamanho da lista restrita de candidatos (l), a regra de ordenação que será aplicada às tarefas, e a atualização da semente do gerador de número aleatório. Quanto ao tamanho da lista restrita, os seguintes valores serão adotados: 2, 3, 4, 5, 6 e n . Isto permitirá trabalhar com regras aproximadamente gulosas (quando $l=2$), ou com busca totalmente aleatória (quando $l=n$).

Quanto à regra de ordenação, ao invés de usar uma regra dinâmica em que, após um determinado sorteio as tarefas remanescentes são reclassificadas, optou-se por utilizar uma regra fixa de ordenação. Desta forma, quando uma tarefa pertencente à lista restrita é escolhida, a próxima tarefa a compor a lista restrita já será conhecida. Seis regras de ordenação das tarefas foram testadas, sendo que cada regra possui dois critérios ordenação. Em caso de empate, após a aplicação do primeiro critério, o segundo critério servirá para desempatar. Se persistir o empate, então as tarefas serão ordenadas pelo seu índice. As regras desenvolvidas foram: (1) Menor Compatibilidade e Menor Instante de Liberação; (2) Menor Compatibilidade e Maior Penalidade; (3) Menor Instante de Liberação e Menor Compatibilidade; (4) Menor Instante de Liberação e Maior Penalidade; (5) Maior Penalidade e Menor Instante de Liberação; (6) Maior Penalidade e Menor Compatibilidade. Quanto à semente, para cada combinação acima descrita (tamanho de lista restrita e regra de ordenação), serão testadas 100 diferentes sementes, perfazendo um total de $6 \times 6 \times 100 = 3.600$ iterações. Na presente implementação, adotou-se um conjunto de elite com tamanho igual a 10.

```

Procedimento GRASP_PathRelinking(maxIterações)
1   LerDados()
2   Para  $k=1, \dots, \text{maxIterações}$  faça
3       Atualiza configuração  $k$ 
4       Para  $j=1, \dots, n$  faça
5           Atualiza lista restrita de candidatos de tamanho  $l(k)$ 
6           Sorteia uma tarefa  $i$  em  $RCL(k)$ 
7           Insere a tarefa  $i$  na melhor posição viável de inserção
8       Fim
9       BuscaLocal()
10      PathRelinking(sol_elite, sol_GRASP)
11      Atualiza ConjuntoElite
12      Atualiza MelhorSolução
13  Fim
14  Retorna MelhorSolução
Fim GRASP

```

Figura 3 – Estrutura básica do GRASP com Path Relinking

A rotina do Path Relinking inicia ajustando as soluções de referências (inicial e guia), dependendo da estratégia escolhida (“forward”, “backward”, “mixed”). A aplicação do Path Relinking consistirá em realizar os movimentos que inserem as tarefas que estão fora de suas rotas correspondentes (na solução guia), na solução inicial. Isto pode ser feito tanto por meio de um movimento de inserção (“relocate”), quanto por meio de um movimento de troca (“swap”) entre duas tarefas. Após todos os possíveis movimentos serem avaliados, o melhor movimento será executado. É importante lembrar que o melhor movimento poderá piorar o valor da função objetivo. Por último, a versão Randomized Path Relinking (RPR) consiste em não necessariamente adotar o melhor movimento possível a partir de um dado estágio do Path Relinking.

6. Testes Computacionais

Foi desenvolvido um gerador de cenários adaptado de Crauwels et al. (1998). Em todos os testes realizados, foi verificado o “gap” das soluções obtidas, o qual é definido como a distância relativa da função objetivo gerada pelo método geração de colunas em relação à função objetivo da melhor solução obtida, calculado da seguinte forma: $GAP = 100(QM - GC)/QM$, onde QM = Função objetivo do método empregado; GC = Função objetivo do método geração de colunas.

Os problemas testes foram gerados da seguinte forma: para cada tarefa j um tempo de processamento p_j inteiro foi gerado empregando-se uma distribuição uniforme entre $[1, 100]$ e, para a penalidade, utilizou-se a faixa $[1, 10]$. O horizonte de planejamento é denotado por $\frac{\sum_{j=1}^n p_j}{v}$ em que v é o número de embarcações. A dificuldade do problema depende do fator RDD (Relative Range of Due

Dates) e do TF (“Tardiness Factor”). Os valores de RDD e TF são {0,2; 0,4; 0,6; 0,8; 1,0}. O instante de liberação é calculado em duas etapas:

$$dp_j = [\sum_{j=1}^n p_j(1 - RDD/2), \sum_{j=1}^n p_j(1 + RDD/2)] \quad (5)$$

$$r_j = \max(0, dp_j - (\min_{k=1}^n \{dp_k\})) \quad (6)$$

Na equação (5) tem-se o cálculo do fator dp_j (instante de liberação preliminar) e, na equação (6) o instante de liberação efetivo. A data de finalização da tarefa (due date) é dada por $d_j = r_j + p_j$. Cada fator TF é combinado com todos os RDDs, o que totaliza 25 instâncias teste. Foram geradas 25 instâncias testes 4 embarcações e 30 tarefas (4mx30n), 5 embarcações e 40 tarefas (5mx40n) e 6 embarcações e 50 tarefas (6mx50n).

6.1. Análise dos Resultados

Os algoritmos foram processados em uma estação de trabalho com processador Intel® Core™ I7 2.80GHz e 16374 MB de memória RAM DDR3 SDRAM para executar o programa codificado em linguagem C++. Os resultados estão apresentados na tabela 1, em que: a coluna 1 apresenta o número da instância; a coluna 2, o valor da relaxação linear da função objetivo, obtida pelo método de geração de colunas; das colunas 3 em diante, apresenta-se o gap, tal qual acima descrito, da aplicação dos seguintes algoritmos: GRASP; GRASP + Forward PR (PR Fw); GRASP + Backward PR (PR Bw); GRASP + Mixed PR (PR Mxd); GRASP + Randomized Forward PR (PR Fw Rnd); GRASP + Randomized Backward PR (PR Bw Rnd); GRASP + Randomized Mixed PR (PR Mxd Rnd).

Tabela 1 – Resultado das instâncias de 30, 40 e 50 tarefas

Instância	Gap % [30 Tarefas x 4 Navios]								Gap % [40 Tarefas x 5 Navios]								Gap % [50 Tarefas x 6 Navios]							
	FOCG	GRASP	PR Fw	PR Bw	PR Mxd	PR Fw Rnd	PR Bw Rnd	PR Mxd Rnd	FOCG	GRASP	PR Fw	PR Bw	PR Mxd	PR Fw Rnd	PR Bw Rnd	PR Mxd Rnd	FOCG	GRASP	PR Fw	PR Bw	PR Mxd	PR Fw Rnd	PR Bw Rnd	PR Mxd Rnd
1	15.417	0,78%	0,73%	0,77%	0,77%	0,73%	0,76%	0,73%	12.354	0,87%	0,87%	0,87%	0,86%	0,86%	0,86%	0,87%	19.685	0,88%	0,87%	0,86%	0,87%	0,89%	0,92%	0,89%
2	9.059	1,53%	1,54%	1,52%	1,54%	1,55%	1,54%	1,54%	11.196	0,80%	0,74%	0,82%	0,75%	0,79%	0,77%	0,75%	11.768	1,43%	1,43%	1,37%	1,42%	1,47%	1,39%	1,40%
3	6.694	2,22%	2,15%	2,15%	2,15%	2,15%	2,15%	2,15%	6.130	2,76%	3,16%	2,76%	2,76%	3,08%	2,99%	2,98%	12.450	1,40%	1,34%	1,14%	1,11%	1,25%	1,22%	1,20%
4	2.107	3,13%	2,59%	2,59%	2,50%	3,61%	2,59%	2,50%	2.698	5,66%	5,33%	4,56%	4,36%	4,60%	5,43%	5,53%	7.817	2,80%	2,89%	2,49%	2,58%	2,64%	2,73%	2,53%
5	1.798	9,10%	9,10%	9,15%	9,10%	9,15%	9,15%	9,33%	789	6,18%	4,48%	4,48%	4,48%	6,07%	6,07%	3,66%	2.498	7,24%	6,83%	5,49%	6,23%	6,69%	6,83%	7,03%
6	6.326	0,28%	0,25%	0,13%	0,13%	0,13%	0,13%	0,14%	9.602	0,40%	0,45%	0,37%	0,37%	0,38%	0,38%	0,38%	22.599	0,69%	0,68%	0,69%	0,68%	0,71%	0,71%	0,71%
7	4.366	2,50%	2,50%	2,50%	2,50%	2,50%	2,50%	2,57%	5.918	1,63%	1,53%	1,20%	1,23%	1,30%	1,60%	1,27%	15.720	0,58%	0,63%	0,59%	0,59%	0,48%	0,48%	0,63%
8	2.309	3,31%	3,31%	3,31%	3,31%	3,31%	3,31%	3,31%	4.538	0,94%	0,94%	0,59%	0,59%	0,94%	0,59%	0,59%	11.165	2,16%	1,70%	1,98%	1,53%	1,68%	1,82%	1,66%
9	3.542	2,93%	2,93%	2,93%	2,93%	2,93%	2,93%	2,93%	3.370	3,69%	3,22%	2,97%	3,16%	3,24%	3,38%	2,99%	9.334	2,54%	2,62%	1,76%	2,16%	2,15%	2,98%	2,15%
10	1.711	3,82%	4,25%	3,82%	3,66%	3,82%	3,22%	4,57%	2.169	8,52%	7,62%	7,74%	6,55%	7,23%	9,21%	6,55%	2.802	3,74%	3,08%	3,58%	2,57%	3,91%	2,88%	2,84%
11	9.481	0,47%	0,47%	0,47%	0,47%	0,47%	0,47%	0,48%	16.600	0,47%	0,47%	0,47%	0,47%	0,49%	0,50%	0,47%	18.094	0,33%	0,32%	0,29%	0,34%	0,33%	0,29%	0,32%
12	8.908	0,69%	0,82%	0,70%	0,76%	0,76%	0,72%	0,60%	8.735	0,95%	1,08%	1,08%	1,01%	1,06%	0,94%	1,14%	15.734	0,87%	0,73%	0,64%	0,64%	0,84%	0,71%	0,76%
13	5.634	3,71%	3,84%	3,97%	3,71%	3,71%	3,84%	3,71%	6.746	2,36%	2,47%	1,89%	2,53%	2,37%	2,53%	2,58%	9.318	2,10%	2,05%	1,93%	2,10%	1,99%	1,90%	2,23%
14	4.388	4,84%	5,02%	5,04%	4,98%	5,06%	4,98%	5,06%	4.918	4,11%	3,64%	3,66%	3,63%	4,10%	3,93%	4,06%	14.299	2,37%	2,42%	2,52%	2,62%	2,62%	2,41%	2,10%
15	3.016	5,34%	5,22%	5,22%	5,10%	5,13%	5,04%	5,04%	5.939	3,67%	3,46%	3,07%	3,51%	3,51%	3,34%	3,46%	7.972	3,05%	2,96%	2,02%	3,28%	3,45%	2,72%	2,83%
16	8.652	1,13%	1,06%	1,06%	1,09%	1,07%	1,07%	1,07%	16.139	1,22%	1,19%	1,19%	1,18%	1,18%	1,19%	1,19%	24.191	0,83%	0,84%	0,83%	0,84%	0,82%	0,84%	0,84%
17	7.408	1,28%	1,25%	1,25%	1,25%	1,27%	1,25%	1,25%	10.847	1,65%	1,65%	1,61%	1,61%	1,65%	1,67%	1,62%	18.571	1,46%	1,52%	1,45%	1,46%	1,52%	1,48%	1,51%
18	7.484	2,36%	2,25%	2,18%	2,21%	2,35%	2,37%	2,25%	9.623	2,30%	2,27%	2,18%	2,18%	2,26%	2,15%	2,15%	17.025	1,29%	1,24%	1,24%	1,23%	1,26%	1,28%	1,32%
19	6.461	2,18%	1,99%	1,99%	2,00%	1,99%	1,99%	1,97%	6.392	2,01%	2,01%	2,01%	2,01%	2,01%	2,01%	2,01%	11.286	2,25%	2,32%	2,08%	2,34%	2,17%	2,44%	2,38%
20	1.601	4,07%	4,07%	4,07%	4,07%	4,07%	4,07%	4,07%	4.379	4,12%	4,16%	4,05%	4,37%	4,35%	4,45%	4,03%	11.832	2,20%	2,26%	2,02%	2,22%	2,29%	2,26%	2,27%
21	8.986	1,48%	1,47%	1,47%	1,47%	1,47%	1,47%	1,47%	10.728	1,69%	1,69%	1,69%	1,69%	1,69%	1,70%	1,69%	21.524	1,13%	1,13%	1,13%	1,13%	1,13%	1,15%	1,15%
22	11.008	1,62%	1,62%	1,62%	1,62%	1,62%	1,62%	1,62%	13.275	1,27%	1,28%	1,27%	1,28%	1,27%	1,27%	1,27%	18.114	1,33%	1,34%	1,32%	1,33%	1,36%	1,35%	1,32%
23	14.882	1,04%	1,04%	1,04%	1,04%	1,04%	1,04%	1,04%	13.592	1,19%	1,19%	1,18%	1,18%	1,19%	1,19%	1,20%	20.472	1,42%	1,41%	1,39%	1,41%	1,40%	1,41%	1,41%
24	8.002	1,47%	1,47%	1,42%	1,42%	1,42%	1,42%	1,42%	10.941	1,60%	1,61%	1,48%	1,65%	1,60%	1,62%	1,59%	20.639	1,14%	1,16%	1,13%	1,14%	1,16%	1,16%	1,15%
25	8.466	1,79%	1,78%	1,78%	1,78%	1,79%	1,78%	1,78%	10.350	2,55%	2,46%	2,46%	2,63%	2,50%	2,50%	2,50%	14.864	1,35%	1,41%	1,43%	1,28%	1,41%	1,43%	1,24%
Média	2.52%	2.51%	2.49%	2.46%	2.52%	2.46%	2.50%		Média	2.50%	2.36%	2.23%	2.24%	2.39%	2.49%	2.26%	Média	1.86%	1.81%	1.65%	1.72%	1.82%	1.79%	1.75%

7. Conclusões e Futuros Trabalhos

A presente pesquisa estudou um problema real de programação de embarcações especializadas em operações de lançamento de “risers”, o qual foi modelado e resolvido como um problema de máquinas paralelas não-relacionadas. Foi adotada a abordagem heurística, sendo testado o GRASP com Path Relinking. Para avaliação dos resultados, foram utilizados limitantes inferiores dados pela versão linear do método de geração de colunas. Foi comprovado que as soluções fornecidas são de boa qualidade, tendo em vista que os gaps ficaram inferiores a 2,5%. A aplicação do Path Relinking ajudou a melhorar a qualidade das soluções. As instâncias mais difíceis de resolver são as com

TF = 0,2 e 0,4. Pretende-se melhorar ainda mais estes resultados, realizando processamento simultâneo das diversas variações do Path Relinking, com recursos multi-threads e aumento na semente de números aleatórios.

8. Agradecimentos

Este trabalho contou com o apoio financeiro parcial da ANP por meio do programa – PRH-ANP/MCT Nº19 e da FINEP através do convênio 01.09.0458.00.

9. Referências

- ALIDAEI, B.; ROSA, D. Scheduling parallel machines to minimize total weighted and unweighted tardiness, *Computers and Operations Research*, 24, p.775-788, 1997.
- BISKUP, D.; HERRMANN, J.; GUPTA, J.N.D. Scheduling identical parallel machines to minimize total tardiness, *International Journal of Production Economics*, 115, p.134-142, 2008.
- BRUCKER, P. *Scheduling Algorithms*, Springer, Berlin, 2007.
- CHENG, T.C.E.; SIN, C.C.S. A state-of-the-art review of parallel-machine scheduling research, *European Journal of Operational Research*, 47, p.271-292, 1990.
- CRAUWELS, H.A.J, POTTS, C.N.; VAN WASSENHOVE, L.N. Local search heuristics for the single total weighted tardiness scheduling problem, *Inform Journal on Computing*, 10, p.341-350, 1998.
- FEO, T.A.; RESENDE, M.G.C. A probabilistic heuristic for a computationally difficult set covering problem, *Operations Research Letter*, 8, p.67-71, 1989.
- FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures, *Journal of Global Optimization*, 6, p.109-133, 1995.
- FULLER, J.A. Optimal solutions versus ‘good’ solutions: an analysis of heuristic decision making, *Omega The International Journal of Management Science*, 6, p.479-484, 1978.
- GLOVER, F. Tabu search and adaptive memory programming – advances, applications and challenges, em Barr, R. S.; Helgason, R.V. e Kennington, J. L. (Eds.), *Interfaces in Computer Science and Operations Research*, Kluwer Academic Publishers, Boston, p.1-75, 1996.
- GLOVER, F.; LAGUNA, M.; MARTÍ, R. Fundamentals of scatter search and path relinking, *Control and Cybernetics*, 29, p.653-684, 2000.
- KOULAMAS, C. Decomposition and hybrid simulated annealing heuristics for the parallel-machine total tardiness problem, *Naval Research Logistics*, 44, p.109-125, 1997.
- LIAW, C-F., LIN, Y-K., CHENG, C-Y.; CHEN, M. Scheduling unrelated parallel machines to minimize total weighted tardiness, *Computers & Operations Research*, 30, p.1777-1789, 2003.
- MORTON, T.E.; PENTICO, D.W. *Heuristic scheduling systems: with applications to production systems and project management*, John Wiley & Sons, New York, 1993.
- PINEDO, M, *Scheduling: theory, algorithm, and systems*, Prentice-Hall, New Jersey, 2002.
- RESENDE, M.G.C.; RIBEIRO, C.C. Grasp with path relinking: recent advances and applications, *AT & T Labs Technical Report TD-5TU726*, p.1-24, 2003.
- RESENDE, M.G.C.; RIBEIRO, C.C.; GLOVER, F.; R. MARTÍ, *Scatter search and path-relinking: Fundamentals, advances, and applications*, em Gendreau, M. e Potvin, J.-Y. (Eds.), *Handbook of Metaheuristics*, Springer, New York, p.87-107, 2010.
- SHIM, S-O.; KIM, Y-D. Scheduling on parallel identical machines to minimize total tardiness, *European Journal of Operational Research*, 177, p.135-146, 2007.
- UNLU, Y.; MASON, S.J. Evaluation of mixed integer formulations for non-preemptive parallel machine scheduling problems, *Computers & Operations Research*, 58, p.785-800, 2010.
- VAN DEN AKKER, J.M.; HURKENS, C.A.J.; SAVELSBERGH, M.W.P. Time-indexed formulations for machine scheduling problem: column generation, *INFORMS Journal on Computing*, 12, p.111-124, 2000.