

PROPOSTA E ANÁLISE EXPERIMENTAL DE HEURÍSTICAS GRASP PARA UM PROBLEMA DE ROTEAMENTO PERIÓDICO DE VEÍCULOS

Luciana Brugiolo Gonçalves, Simone de Lima Martins e Luiz Satoru Ochi

Instituto de Computação – Universidade Federal Fluminense (UFF)
Passos da Pátria – Niterói – RJ – Brasil

{lgoncalves, simone, satoru}@ic.uff.br

RESUMO:

Neste trabalho são apresentadas propostas para determinar soluções aproximadas para uma variante do problema de roteamento periódico de veículos (PRPV), o problema de roteamento periódico de Unidades Móveis de Pistoneio (PRP-UMP). Este problema difere do PRPV em vários aspectos. A principal diferença, responsável por aumentar a complexidade do problema, é que o número de visitas requeridas por um cliente durante o período de planejamento não é previamente estabelecido. O PRP-UMP modela uma aplicação real associada à exploração de poços petrolíferos, situação comum na região nordeste do Brasil. Neste trabalho é apresentada uma formulação matemática mista que descreve o PRP-UMP como um problema de programação linear e também versões puras e híbridas da heurística GRASP. Resultados experimentais ilustram a eficácia do GRASP com memória adaptativa quando comparado com as versões puras.

Palavras chave: Roteamento periódico, metaheurística, memória adaptativa.

ABSTRACT

In this paper, we present some proposals to approximately solve a version of the period vehicle routing problem (PVRP), the Period Bump Mobile Units Routing Problem (PBMURP). This problem differs from the well-known PVRP in several aspects. One major difference between them, responsible for increasing the complexity of the problem, is that the number of visits required by a customer during the period is not previously determined. The PBMURP models a real application found in the Northeastern part of Brazil concerning the exploitation of oil in onshore oil wells. In this work, we propose a mathematical formulation describing the PBMURP as an mixed linear programming problem and GRASP heuristics including pure and hybrid versions. Experimental results illustrate the effectiveness of GRASP with adaptive memory over pure GRASP.

Keywords: period vehicle routing problem, metaheuristics, adaptive memory

1. Introdução

O modelo básico de Problema de Roteamento de Veículos, aqui denotado por PRV, tem como objetivo minimizar os custos de transporte relacionados ao atendimento de um conjunto de clientes a partir de uma frota de veículos homogêneos localizados num depósito origem. A solução deste modelo básico consiste em gerar um conjunto de rotas (uma para cada veículo) com início e término no depósito de modo que a demanda de cada um dos clientes seja atendida por um único veículo e em uma única visita respeitando as restrições de capacidade dos veículos.

A literatura apresenta um elevado número de variações do PRV modelando diferentes aplicações. Em algumas destas não é necessário estabelecer rotas que contenham todos os clientes envolvidos, apenas um subconjunto faz parte da solução do problema, como por exemplo *Orienteering Problem* e *Traveling Purchase*. Uma outra variante do PRV muito abordada pela literatura afim é conhecida como Problema de Roteamento Periódico de Veículos (Christofides e Beasley (1984)), onde o objetivo é planejar visitas para os clientes num dado horizonte de planejamento.

Neste trabalho é abordado um modelo real de roteamento de veículos englobando características destes dois modelos, ou seja, onde uma solução apresenta um planejamento de visitas em um determinado período, sendo desnecessário visitar todos os clientes disponíveis em cada um dos dias. O modelo real focado está associado a um problema de extração de derivados de petróleo de poços terrestres da região nordeste do Brasil (Aloise *et al.* (2004)).

Para resolver este problema, que é detalhado na Seção 2, uma formulação matemática, que descreve o PRP-UMP como um problema de programação linear mista é apresentada na Seção 3 e heurísticas são descritas na Seção 4. Resultados computacionais são apresentados na Seção 5 e conclusões são discutidas na Seção 6.

2. O Problema de Roteamento Periódico de Unidades Móveis de Pistoneio

Nos campos petrolíferos terrestres existe um conjunto de poços que não têm capacidade própria de elevação de óleo até a superfície, referenciado como poços não surgentes. Nestes poços há necessidade de utilização de mecanismos artificiais para a extração do fluido, tais mecanismos que podem ser divididos em fixos e móveis. Quando um poço não apresenta vazão suficiente que justifique o custo da utilização de métodos fixos a utilização dos mecanismos móveis, Unidade Móvel de Pistoneio (UMP), por exemplo, apresenta-se como uma boa estratégia para a solução do problema, como pode ser observado em Aloise *et al.* (2004).

No uso da UMP, o fluido coletado é inicialmente armazenado na própria unidade e posteriormente transferido para a estação de tratamento de óleo - ETO. Tal estação é o local de onde toda UMP deve partir e para onde deve retornar após coletar o óleo de um conjunto de poços, sempre respeitando um tempo máximo de duração da rota que é determinado pela jornada de trabalho. Com a capacidade do tanque da UMP é limitada, utiliza-se um caminhão tanque como apoio para realizar o transporte do óleo, deixando assim a UMP com a tarefa exclusiva de coletar o fluido dos poços. Por este motivo, é desnecessário considerar a capacidade da UMP como restrição para o problema.

Após a retirada de fluido, o poço por hora vazio, começa um processo de reenchimento onde o nível do óleo irá aumentar até atingir o nível estático. Desta forma, é necessário aguardar um intervalo de tempo (tempo de reenchimento) para que uma nova coleta seja realizada.

No Problema de Roteamento Periódico de Unidades Móveis de Pistoneio (PRP-UMP), dentro de um horizonte de planejamento deve-se definir, para cada dia, os poços que cada UMP deve pistonear, bem como a ordem de visitação dos poços, o que determina uma rota. Cada rota é definida no subconjunto de poços que já tenham alcançado seu nível estático para o dia em questão. Desta forma, é possível que alguns poços não sejam atendidos durante o período ou que não sejam visitados o número máximo possível de vezes. Isto pode ocorrer na prática devido ao elevado número de poços e ao número reduzido de UMP disponíveis.

A diferença entre este modelo e o PRPV, que aumenta consideravelmente a

complexidade do problema, se refere ao intervalo de atendimento dos poços. No PRPV as possíveis combinações de visitas são pré-determinados enquanto no modelo considerado neste trabalho apenas um intervalo mínimo entre duas coletas é estabelecido.

O PRP-UMP foi proposto em Gonçalves *et al.* (2005). Na literatura existem outras abordagens referentes ao roteamento de UMP, mas para o período referente a um único dia. O objetivo do PRP-UMP é gerar rotas diárias para cada UMP de modo a maximizar a quantidade de fluido coletado durante todo o período de planejamento respeitando as restrições envolvidas, como jornada de trabalho e intervalo de reenchimento dos poços. Na próxima seção é apresentada uma formulação matemática que descrever o PRP-UMP como um problema de programação inteira mista.

3. Formulação Matemática

Considere um grafo não direcionado $G=(V,E)$, onde V representa o conjunto de poços acrescido da Estação de Tratamento de Óleo (ETO), com $|V| = n$, e E o conjunto de arestas (u,v) às quais encontram-se associados t_{uv} que indicam o tempo de percurso entre os poços u e v . A cada poço i pertencente ao conjunto \hat{V} , onde $V = \hat{V} \cup \{0\}$, temos associado v_i igual a vazão esperada do poço i , s_i representando o tempo de permanência¹ no poço i e g_i correspondendo ao tempo de reenchimento do poço i . O PRP-UMP consiste em construir, para o conjunto D de dias, onde $|D|=p$, rotas com duração máxima de $TMAX$ unidades de tempo para cada uma das r UMP do conjunto R de veículos.

Para esta formulação são definidas três variáveis: x_{it} , que assume valor 1 se o poço i é visitado no dia t , 0 caso contrário. A variável z_{ijkt} , que indica com valor 1 que o poço j é visitado logo após o poço i pela UMP k no dia t , e 0 caso contrário. Por fim, temos a variável w_{ijkt} que representa quantidade de óleo transportado no percurso (i,j) pela UMP k no dia t .

Função Objetivo:

$$\text{Maximizar } F = \sum_{t=1}^p \sum_{i=1}^n v_i \times x_{it} \quad (1)$$

Restrições:

$$\sum_{i=0}^n \sum_{j=0, j \neq i}^n (t_{ij} + s_i) \times z_{ijkt} \leq TMax \quad \forall t \in D, \forall k \in R \quad (2)$$

$$\sum_{k=1}^r \sum_{i=0, i \neq q}^n z_{iqkt} = x_{qt} \quad \forall q \in \hat{V}, \forall t \in D; \quad (3)$$

$$\sum_{k=1}^r \sum_{i=0, i \neq q}^n z_{qikt} = x_{qt} \quad \forall q \in \hat{V}, \forall t \in D; \quad (4)$$

$$\sum_{i=0, i \neq q}^n z_{iqkt} - \sum_{i=0, i \neq q}^n z_{qikt} = 0 \quad \forall q \in \hat{V}, \forall t \in D, \forall k \in R; \quad (5)$$

$$\sum_{k=1}^r z_{ijkt} \leq \frac{x_{it} + x_{jt}}{2} \quad \forall i, j \in \hat{V} | i \neq j, \forall t \in D; \quad (6)$$

$$x_{0t} = 1 \quad \forall t \in D; \quad (7)$$

$$\sum_{i=1}^n z_{i0kt} = 1 \quad \forall t \in D, \forall k \in R; \quad (8)$$

$$\sum_{j=0, j \neq q}^n w_{qjkt} = \sum_{i=0, i \neq q}^n (w_{iqkt} + v_q \times z_{iqkt}) \quad \forall k \in R, \forall t \in D, \forall q \in \hat{V}; \quad (9)$$

$$w_{ojkt} = z_{ojkt} \quad \forall j \in \hat{V}, \forall k \in R, \forall t \in D; \quad (10)$$

¹ Tempo necessário para instalação e desinstalação da UMP acrescido do tempo de coleta.

$$w_{ijkt} \geq z_{ijkt} \quad \forall i, j \in V \mid i \neq j, \forall k \in R, \forall t \in D; \quad (11)$$

$$z_{ijkt} \geq \frac{w_{ijkt}}{\sum_{j=1}^n v_i} \quad \forall i, j \in V \mid i \neq j, \forall k \in R, \forall t \in D; \quad (12)$$

$$\sum_{t=t'+1}^{t'+g_i} x_{it} \leq 1 - x_{it'} \quad \forall t' \in D, \forall i \in \hat{V} \quad (13)$$

$$x_{it} \in \{0,1\} \quad \forall i \in V, \forall t \in D; \quad (14)$$

$$y_{ijkt} \in \{0,1\} \quad \forall i, j \in V, \forall t \in D, \forall k \in R; \quad (15)$$

$$w_{ijkt} \geq 0 \quad \forall i, j \in V, \forall t \in D, \forall k \in R; \quad (16)$$

Na função objetivo do problema (Equação 1) almeja-se maximizar a vazão coletada durante todo o período por todas as Unidades Móveis de Pistoneio. As desigualdades (2) garantem que o somatório dos tempos gastos em cada rota não exceda o limite da jornada diária. As restrições (3), (4) e (5) impõem que a seleção de um poço i para pistoneio implica que uma, e somente uma UMP deve chegar, pistonear e partir desse poço a cada visita. As restrições (6) garantem que, dados dois poços i e j marcados para pistoneio em um determinado dia t , caso a coleta seja feita na seqüência j logo após i , apenas um veículo poderá fazê-la.

Em (7), certifica-se que a ETO deve estar presente em cada rota em todos os dias. Nas restrições (8) se estabelece que não sejam geradas rotas vazias, ou seja, cada UMP deve coletar o óleo de pelo menos um poço a cada dia.

As restrições de (9) a (10) garantem que as rotas geradas não sejam desconexas da origem. Para isso, uma variável de fluxo (w_{ijkt}) é associada a cada aresta (i,j) para cada veículo em cada dia. Em (9) temos que a quantidade de fluxo que sai de um poço é igual a quantidade de fluxo que entra somada a vazão associada ao poço. Para garantir a corretude do modelo, foi associado um fluxo unitário a todos os arcos utilizados pelas UMP's para sair da ETO (restrição (10)) fluxo esse que não é computado pela função objetivo. Pelas restrições (11), pode-se observar que se um veículo faz uso de uma determinada aresta em um determinado dia ($z_{ijkt} = 1$), então haverá algum fluxo positivo associado a este uso ($w_{ijkt} \geq 1$). As restrições (12) confirmam que numa aresta, quando há um fluxo positivo, necessariamente um veículo deve passar pela aresta no dia do período correspondente ($z_{ijkt} = 1$).

As restrições (13) impedem que os poços sejam pistoneados em intervalos que não permitam o reenchimento dos mesmos. Finalmente, as demais restrições (14) a (16) determinam o domínio das variáveis do problema.

Devido a elevada complexidade do problema (*NP-Hard*), a solução pelo método exato só é possível na prática para instâncias de pequeno porte. Diante desta limitação, na próxima seção, são apresentados algoritmos heurísticos para a solução aproximada do PRP-UMP.

4. Heurísticas

A metaheurística GRASP proposta por Feo e Resende (1995) é um processo iterativo, onde cada iteração consiste de duas fases: construção e busca local. Na fase de construção uma solução viável é construída, e na etapa seguinte sua vizinhança é explorada pela busca local. O resultado é a melhor solução encontrada de todas iterações. Nas próximas subseções são descritas as heurísticas utilizadas nos GRASP propostos para o PRP-UMP.

4.1 Heurísticas de Construção

Foram desenvolvidas duas heurísticas de construção aplicadas nos algoritmos GRASP propostos. A primeira baseada no Método das Pétalas (C-MP), e a segunda na heurística de Inserção Mais Próxima (C-IMP). Nas duas heurísticas, as rotas são geradas de forma seqüencial, ou seja, a cada

dia do período de planejamento, uma rota é construída para cada UMP.

Nas heurísticas de construção propostas, deseja-se gerar, para cada dia do período, uma rota para cada uma das UMP, de forma que a duração não seja maior que a jornada de trabalho pré-estabelecida e a maior quantidade de óleo seja coletada. Uma solução do PRP-UMP é composta por um conjunto S de $|R| \times |D|$ rotas, onde R é o conjunto de UMP e D o conjunto de dias, sendo associado a cada rota um custo igual ao tempo de duração das mesmas.

O funcionamento das heurísticas pode ser descrito da seguinte forma: Para cada dia, verificam-se os poços disponíveis aptos a participarem das rotas naquele dia que vão compor a lista de candidatos (LC). Depois disso uma rota é gerada para cada UMP. Para selecionar o primeiro poço a integrar a rota, a LC é ordenada segundo uma função de avaliação, onde os poços melhor avaliados são selecionados para compor a LRC (Lista Restrita de Candidatos). A função de avaliação relaciona a vazão dos poços com a distância destes à ETO. Depois de determinar a LRC, um dos poços que a compõem é escolhido aleatoriamente. A rota inicial é construída, sendo associado a esta um custo igual ao tempo de permanência da UMP para atender o poço, acrescido do tempo necessário para ir e voltar ao mesmo.

Após a inserção do poço inicial na rota, o processo prossegue até que, na tentativa de uma nova inserção, a duração da rota ultrapasse a jornada. Para selecionar os próximos poços que farão parte da rota, a LRC é atualizada de acordo com a heurística utilizada (MP ou IMP). Em seguida, um poço é sorteado. Se for possível inserir este poço, a rota e seu custo são atualizados, caso contrário a rota é concluída.

Na heurística de construção C-MP, um procedimento inspirado no método da pétala é utilizado para determinar quais poços farão parte da LRC, de onde será selecionado o poço que irá compor a rota. Sempre que for necessário selecionar um novo poço, uma única pétala é gerada considerando a ETO como ponto base (p_b) e o último poço inserido na rota como ponto de referência (p_r), sendo o tamanho da pétala limitado pelo número de poços que integrarão a LRC. Depois de gerada a pétala, os poços que a compõem são avaliados segundo a função de custo $f(i) = v_i / t_{ij}$ (com $j = p_r$). A escolha do poço é feita através da aplicação do método da roleta, pelo qual a probabilidade de um poço ser escolhido é proporcional ao custo de $f(i)$. Caso seja possível realizar a inserção sem que a duração da rota ultrapasse a jornada de trabalho, a inserção é efetuada e o ponto de referência é atualizado (p_r poço inserido) para geração da próxima pétala (atualização da LRC). Caso contrário, o poço não é inserido e a rota é finalizada fazendo com que UMP retorne ao ponto de partida (ETO).

Já a heurística C-IMP, tendo como ponto de partida a rota contendo um único poço, para selecionar o próximo poço a ser inserido, os k poços considerados mais atrativos para cada poço pertencente à mesma são avaliados de acordo com a função $f(i) = v_i / t_{ij}$, onde j representa o poço que está na rota. Estes poços são dispostos em ordem não crescente em relação a esta função e o próximo poço a fazer parte da rota é selecionado aleatoriamente entre aqueles com avaliação superior ao limite $l = \alpha \times \min(f(i)) + (1 - \alpha) \times \max(f(i))$.

4.2 Heurísticas de Busca Local

Para etapa de busca local dois procedimentos são propostos. O primeiro analisa cada dia do período de forma independente (Busca Local Diária - BLD) e o segundo trata o período de forma global (Busca Local Periódica - BLP).

O objetivo da BLD é conseguir uma melhor distribuição, nas rotas das diferentes UMP's, dos poços alocados para visitaç o em um determinado dia, bem como adicionar novos poços que possivelmente estejam dispon veis no momento, tratando individualmente cada dia do per odo. Para tanto   determinada a rota que melhor atende a cada poço com um menor custo, entendendo custo como a soma da duraç o de todas as rotas. Ap s a realocaç o dos poços, verifica-se a possibilidade de inserç o de novos poços caso tenha ocorrido reduç o na duraç o das rotas.

J  a BLP prop e alterar os dias de visitaç o daqueles poços que podem ser melhor

explorados devido a sua boa vazão e ao reduzido número de coletas realizadas a este poço na solução atual. Para isso, determina-se o desperdício² em relação à vazão de cada poço. Um subconjunto dos poços que estão com maior desperdício tem seus dias de visitação alterados. Então, são determinados novos conjuntos de dias de visitação que explorem ao máximo a vazão dos poços selecionados, reduzindo o desperdício total.

4.3 Reconexão de Caminhos

Na versão inicial da metaheurística GRASP, cada iteração é realizada de forma independente. Este comportamento impossibilita que as melhores características obtidas a cada iteração sejam reaproveitadas nas iterações posteriores. Uma forma de melhorar o desempenho do GRASP é a utilização de buscas intensivas numa região promissora fazendo uso de informações relevantes obtidas em iterações passadas deste algoritmo. Neste sentido, uma abordagem que vem sendo utilizada com sucesso é a Reconexão de Caminhos (RC).

Esta técnica, proposta originalmente por Glover (1996), consiste basicamente em explorar as soluções intermediárias entre duas soluções. Para tanto, definidas as soluções origem s_0 e destino s_d , para gerar cada solução intermediária, movimentos são selecionados de forma a introduzir na solução corrente (inicialmente s_0), atributos presentes na solução destino (s_d). Explorar este tipo de característica vem se mostrando uma estratégia muito promissora, como pode ser observado em Bastos *et al.* (2005), Silva *et al.* (2007) e Pinto *et al.* (2006).

Na sua forma mais básica o processo pode ser visto da seguinte forma: Num primeiro passo, as diferenças entre as duas soluções são analisadas de forma a gerar um conjunto Δ , definindo os possíveis movimentos que podem ser executados. Dentre todos os movimentos pertencentes ao conjunto Δ , aquele que implica em maior ganho ao alterar a solução origem é aplicado, gerando uma nova solução intermediária s_{rc} , sendo então o conjunto Δ atualizado pela remoção do movimento efetuado. Iterativamente, são aplicados movimentos sobre a solução corrente s_{rc} até que a solução destino seja alcançada, isto é, $\Delta = \emptyset$. A melhor solução encontrada durante este processo é tida como resultado desta técnica.

Objetivando-se tirar proveito das melhores características das soluções obtidas em cada iteração do GRASP, neste trabalho foi desenvolvido um módulo de Reconexão de Caminhos para o PRP-UIMP. Para determinar o conjunto Δ , duas soluções do PRP-UIMP são comparadas levando em consideração todas as rotas de cada dia do período.

Na Figura 1 são apresentadas duas soluções s_0 e s_d para um problema que possui dezesseite poços, duas UMP e com período composto por seis dias. Pode-se observar que não há nenhum dia em que todas as rotas sejam iguais, assim, o conjunto de atributos diferentes para s_0 e s_d é composto por cada dia do período (tamanho máximo $\Delta = |D|$), ou seja, $\{d_1, d_2, d_3, d_4, d_5, d_6\}$.

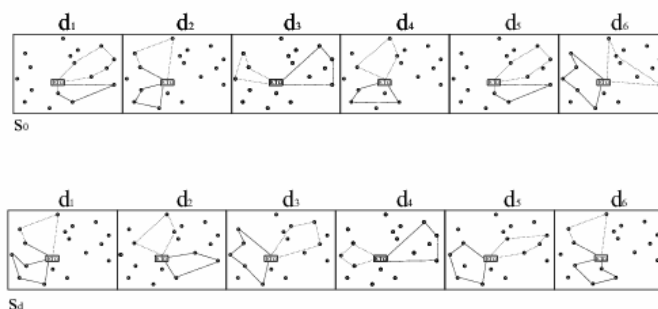


Figura 1 – Reconexão de Caminhos

² Quantidade máxima de óleo possível de ser extraída de um poço durante o período, subtraída da quantidade de óleo que está sendo coletada na solução atual.

Para fazer com que a solução s_0 seja alterada com a adição de características de s_d , a cada etapa as rotas referentes a um determinado dia da solução s_0 são substituídas pelas rotas equivalentes da solução destino. Este tipo de movimento garante que não haverá inviabilidade entre as rotas de um mesmo dia, mas não garante que sejam respeitadas as restrições referentes ao tempo de reenchimento de cada poço. Por isso, após cada movimento é necessário viabilizar a solução.

O procedimento de Reconexão de Caminhos é realizado entre uma solução que acaba de ser gerada e uma outra solução selecionada de um conjunto elite, S_{elite} . Conjunto este onde estão armazenadas as melhores soluções geradas pela heurística até o momento. A solução resultante deste procedimento substituirá uma solução do conjunto S_{elite} se a qualidade desta for superior a qualidade de alguma solução deste conjunto.

Na próxima seção são apresentados os resultados obtidos combinando as heurísticas apresentadas neste trabalho e também resultados da utilização da formulação matemática proposta.

5. Resultados Computacionais

Para avaliar os algoritmos propostos foram realizados experimentos em máquinas Intel Pentium 4 2.8GHz com 512MB de RAM, onde os algoritmos foram implementados usando a linguagem C, compilada no gcc versão 3.3.3 rodando sobre Linux Fedora Core 2.

Para avaliar as heurísticas propostas, quatro configurações GRASP foram testadas: G1 (C-MP +BLD + BLP), G2 (C-MP + BLP + BLD), G3 (C-IMP + BLD + BLP) e G4 (C-IMP +BLP + BLD). Pode ser observado que, em todas as versões, as duas buscas locais são executadas, diferindo somente quanto à ordem em que são aplicadas. Para cada versão GRASP, verificou-se o efeito da utilização do módulo Reconexão de Caminhos.

Dois grupos de instâncias foram utilizados nos testes computacionais. O primeiro grupo com instâncias pequenas onde o número de dias do período varia de 1 a 3, o número de UMP pode ser 1 ou 2 e o número de poços está no intervalo [5,15]. Grupo este que foi utilizado para verificar o desempenho das abordagens em relação ao método exato. O segundo conjunto de instâncias consiste de 17 instâncias onde o número de dias do período se encontra no intervalo [5,15]; o número de UMP no conjunto {2,3}; e o número de poços foi distribuído no intervalo [50,1000]. Estas foram as instâncias usadas nos testes das heurísticas.

5.1 Comparação com a Solução Exata

Para verificação dos resultados das abordagens propostas em relação à execução de um método exato foi utilizado o primeiro conjunto de instâncias testes. Para resolver o modelo matemático foi utilizada a ferramenta CPLEX 10. Na Tabela 1, a primeira coluna identifica estas instâncias, onde o primeiro número indica o número de poços, o segundo corresponde ao número de dias do período de planejamento e o terceiro indica o número de UMP. A segunda coluna mostra o resultado obtido executando-se a ferramenta, tanto no que se refere ao custo da solução como em relação ao tempo de processamento medido em segundos. Da quarta à sétima coluna são apresentados os resultados, em relação ao tempo computacional, exigidos pelas quatro versões G1, G2, G3 e G4 para atingirem a solução ótima. O custo da solução não é apresentado, pois todas as versões GRASP alcançaram o valor ótimo para todas as instâncias. Devido à simplicidade das instâncias e ao reduzido intervalo de tempo gasto para determinar a solução ótima, o módulo de Reconexão de Caminhos não foi utilizado para estas instâncias.

Cada heurística GRASP foi executada cinco vezes para cada instância, sendo apresentado na tabela o tempo médio de todas as execuções. Para este tipo de teste o critério de parada do GRASP foi atingir o valor ótimo.

Tabela1 – Comparação resultado exato com o resultado da metaheurística

Instância	CPLEX		G1	G2	G3	G4
	custo	tempo (s)	tempo (s)	tempo (s)	tempo (s)	tempo (s)
I 05 - 1 - 1	114,00	0,06	0,01	0,00	0,00	0,00
I 10 - 1 - 1	450,00	0,43	0,01	0,01	0,01	0,01
I 10 - 2 - 1	900,00	1,51	0,02	0,03	0,03	0,03
I 10 - 2 - 2	1.300,00	216,38	0,03	0,03	0,04	0,03
I 13 - 1 - 1	600,00	2,72	0,02	0,02	0,03	0,03
I 13 - 2 - 1	1.110,00	148,53	0,04	0,04	0,04	0,04
I 13 - 2 - 2	1.610,00	823,80	0,05	0,05	0,06	0,06
I 14 - 1 - 1	300,00	0,22	0,02	0,01	0,02	0,02
I 14 - 1 - 2	550,00	3,10	0,03	0,03	0,03	0,03
I 14 - 2 - 2	1.050,00	123,93	0,05	0,05	0,05	0,05
I 14 - 3 - 2	1.350,00	3.294,11	0,05	0,07	0,05	0,04
I 15 - 3 - 1	620,00	4.366,10	0,02	0,02	0,02	0,06
Média		748,40	0,03	0,30	0,30	0,30

5.2 Comparação das Heurísticas

Testes preliminares foram executados com o intuito de definir, de forma empírica, os melhores parâmetros a serem utilizados para avaliação dos algoritmos. Assim, são assumidos os seguintes valores: $\alpha = 30\%$, número máximo de iterações 200 e conjunto elite com 4 soluções.

Na Tabela 2 são apresentadas as características das instâncias utilizadas para avaliação das abordagens propostas. Na primeira coluna é apresentado o identificador da instância e na coluna seguinte o número de poços da instância, seguido pelo número de dias do período analisado e pelo número de veículos disponíveis.

Tabela 2 – Descrição das Instâncias

Id	n° poços	período	UMP
Inst 1	50	10	2
Inst 2	50	7	2
Inst 3	50	7	2
Inst 4	50	10	2
Inst 5	50	5	2
Inst 6	100	15	3
Inst 7	100	10	3
Inst 8	150	14	2
Inst 9	150	7	2

Id	n° poços	período	UMP
Inst 10	200	10	2
Inst 11	250	10	3
Inst 12	250	10	3
Inst 13	400	10	3
Inst 14	500	15	3
Inst 15	700	10	3
Inst 16	800	10	2
Inst 17	1000	10	2

Para verificação do desempenho dos algoritmos analisados, cada versão GRASP foi executada 30 vezes para cada uma das 17 instâncias. Uma comparação entre o melhor valor conhecido e a média da qualidade das soluções obtidas nas execuções para cada uma das instâncias é apresentada na Tabela 3, colunas de 2 a 9. Nesta tabela pode-se observar que, entre os resultados obtidos pelas versões G1, G2, G3 e G4, o melhor resultado médio foi obtido por G4 seguido por G3 (ambas utilizando construção IMP, sendo que a versão G4 executa a BLP antes da BLD). Incluindo o módulo de Reconexão de Caminhos (RC) é possível observar melhoria na qualidade de cada um dos algoritmos em relação à sua versão pura.

Na Tabela 4 são ilustradas os ganhos, em percentual, obtidas em cada GRASP puro ao se incluir o módulo de RC. A última linha da tabela mostra o ganho médio percentual pela aplicação da RC. Por esta linha pode-se observar que a contribuição do RC foi muito maior para os casos onde o GRASP puro utiliza a construção por pétalas (G1 e G2).

Tabela 3 – Qualidade da Solução - Percentual em relação ao melhor conhecido.

	G1	G1+RC	G2	G2+RC	G3	G3+RC	G4	G4+RC
Inst 1	0,842	0,905	0,846	0,905	0,938	0,968	0,943	0,972
Inst 2	0,819	0,881	0,824	0,878	0,951	0,974	0,953	0,974
Inst 3	0,858	0,887	0,862	0,887	0,965	0,971	0,968	0,972
Inst 4	0,840	0,852	0,841	0,852	0,976	0,982	0,977	0,982
Inst 5	0,897	0,941	0,894	0,943	0,964	0,976	0,964	0,978
Inst 6	0,880	0,887	0,883	0,889	0,986	0,987	0,989	0,990
Inst 7	0,861	0,866	0,863	0,866	0,976	0,977	0,980	0,990
Inst 8	0,822	0,909	0,817	0,901	0,971	0,982	0,973	0,983
Inst 9	0,799	0,897	0,795	0,896	0,956	0,974	0,957	0,972
Inst 10	0,811	0,894	0,797	0,878	0,963	0,974	0,967	0,976
Inst 11	0,792	0,859	0,776	0,852	0,980	0,980	0,983	0,984
Inst 12	0,797	0,868	0,775	0,861	0,988	0,989	0,991	0,992
Inst 13	0,813	0,882	0,793	0,856	0,987	0,988	0,985	0,985
Inst 14	0,868	0,872	0,863	0,871	0,787	0,888	0,994	0,994
Inst 15	0,854	0,909	0,834	0,881	0,989	0,993	0,986	0,989
Inst 16	0,794	0,910	0,764	0,889	0,931	0,967	0,926	0,969
Inst 17	0,750	0,906	0,730	0,900	0,940	0,986	0,931	0,983
Média	0,829	0,890	0,821	0,883	0,956	0,974	0,969	0,982

Sabe-se que, em geral, o uso de módulos adicionais, como a Reconexão de Caminhos (RC), tende a provocar um acréscimo significativo no tempo total de processamento do algoritmo. O impacto do uso deste módulo é ilustrado na Tabela 5, que apresenta os tempos médios de cada versão GRASP com e sem RC. Os resultados mostram que, mesmo tendo sido chamado em todas as iterações GRASP, a RC não provoca um acréscimo significativo na duração total do GRASP.

Tabela 4 – Ganho percentual na qualidade da solução devido aplicação da RC

	G1	G2	G3	G4
Inst 1	7,51	6,95	3,25	3,08
Inst 2	7,56	6,59	2,46	2,22
Inst 3	3,38	2,91	0,67	0,49
Inst 4	1,40	1,35	0,56	0,47
Inst 5	4,90	5,44	1,29	1,37
Inst 6	0,82	0,65	0,10	0,10
Inst 7	0,60	0,32	0,07	1,04
Inst 8	10,63	10,28	1,14	0,96
Inst 9	12,30	12,73	1,92	1,66

	G1	G2	G3	G4
Inst 10	10,21	10,18	1,09	1,02
Inst 11	8,51	9,82	0,02	0,02
Inst 12	8,96	11,13	0,10	0,08
Inst 13	8,52	7,92	0,07	0,00
Inst 14	0,51	0,92	12,88	0,00
Inst 15	6,45	5,59	0,36	0,34
Inst 16	14,68	16,38	3,91	4,72
Inst 17	20,80	23,28	4,89	5,62
Média	7,51	7,79	2,05	1,37

5.3 Análise Probabilística

Nesta seção, é colocada como meta a análise da robustez das heurísticas aqui propostas. Isso se torna importante na medida que heurísticas com componentes aleatórias, como é o caso do GRASP, podem apresentar soluções bem distintas a cada execução de uma dada instância. Desta forma, foi utilizada uma outra abordagem para análise de desempenho empírico de metaheurísticas, proposta por Aiex *et al.* (2002). Este tipo de análise consiste em tomar os tempos de processamento que cada algoritmo requer para atingir um valor alvo na função objetivo. Para tanto, no instante em que cada algoritmo encontra uma solução melhor ou igual (em problemas de maximização) ao valor alvo fixado, o tempo é registrado e a execução é interrompida.

Para esta bateria de testes, cada algoritmo foi executado 100 vezes para cada instância avaliada usando os mesmos parâmetros inicialmente adotados nas análises anteriores. Após 100 execuções, os tempos foram tomados e dispostos em ordem crescente numa lista L . A cada tempo

Tabela 5 – Média do tempo de CPU em segundos

Inst	G1	G1 + RC	G2	G2 + RC	G3	G3 + RC	G4	G4 + RC
1	3,84	4,48	3,75	4,39	4,66	5,31	4,33	5,00
2	3,81	4,10	4,14	4,43	5,78	6,08	5,08	5,38
3	4,19	4,42	4,34	4,56	6,13	6,35	5,46	5,68
4	4,42	4,90	4,56	5,02	6,62	7,04	5,72	6,15
5	2,25	2,38	2,14	2,28	2,47	2,62	2,13	2,27
6	45,91	49,19	47,79	51,08	57,52	60,97	53,82	57,25
7	15,65	16,72	16,63	17,73	20,14	21,26	18,77	19,89
8	88,92	92,94	86,60	90,53	100,07	104,47	97,08	101,35
9	34,83	35,73	35,00	35,93	40,82	41,79	38,08	39,06
10	124,38	126,92	122,65	125,23	141,97	144,83	138,98	141,83
11	184,58	187,40	183,72	186,59	225,48	228,49	220,45	223,48
12	152,30	155,39	154,37	157,50	218,61	222,16	221,19	224,77
13	497,68	502,68	328,36	331,74	1.060,58	1.065,86	733,79	737,36
14	1.980,75	1.994,37	2.261,20	2.274,58	13.373,46	13.391,67	12.186,41	12.205,68
15	4.109,28	4.119,22	2.922,87	2.929,95	8.869,12	8.880,22	6.574,11	6.581,91
16	6.664,71	6.674,70	7.099,63	7.109,63	9.711,30	9.724,06	9.153,86	9.166,90
17	6.086,22	6.102,49	7.821,34	7.841,50	9.411,04	9.433,12	9.161,23	9.183,08
Média	1.176,69	1.181,06	1.241,12	1.245,45	2.544,46	2.549,78	2.271,79	2.276,88

de CPU t obtido, foi associada a probabilidade $p_i = (i - 1/2)/100$, onde i é a ordem que t aparece na lista ordenada L . A plotagem foi feita tomando-se cada ponto (t_i, p_i) . Caso o algoritmo não tenha sucesso em encontrar o alvo durante 200 iterações, foram concedidas mais 100 iterações para que o mesmo tentasse alcançá-lo. Caso, mesmo após as 300 iterações, o alvo não fosse obtido, considera-se que, nesta execução, o algoritmo não foi capaz de atingir o mesmo.

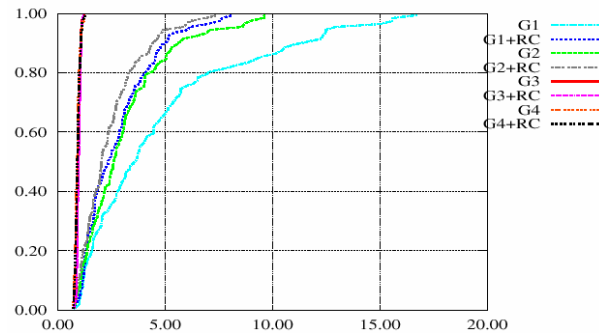
Inicialmente, para se determinar os alvos a serem usados, cada algoritmo proposto foi executado dez vezes com sementes diferentes por 200 iterações. Para cada versão GRASP, calculou-se o resultado médio atingido para cada uma das instâncias. O alvo fácil é assumido como a média alcançada pela versão com pior desempenho, o alvo difícil corresponde a 95% da média atingida pela versão GRASP de melhor desempenho. Como o comportamento para todas as instâncias avaliadas foi parecido, apenas a análise da instância 6 é apresentada.

Na Figura 2 (a) é apresentado o resultado da instância 6 para um alvo fácil. Pode-se perceber que as versões G3 e G4 com e sem RC têm convergência empírica mais rápida se comparadas às versões G1 e G2. Além disso, observa-se que o impacto da RC no desempenho do algoritmo G1 é muito mais significativo que para qualquer uma das abordagens, sendo que, quando G1+RC converge com 100% de acerto demandando um tempo inferior a 10 segundos, para esta mesma convergência, a versão sem RC precisa de mais de 15 segundos. No entanto, a eficiência do módulo de RC não foi suficiente para que as abordagens que utilizam construção por pétala convergissem para o alvo difícil (Figura 2 - b). Pode-se perceber notadamente que as versões G3 e G4, quando utilizam RC têm muito mais chance de convergir em um tempo mais cedo.

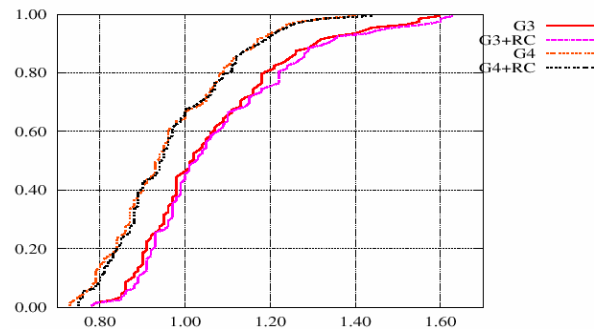
6. Conclusões e Trabalhos Futuros

Neste trabalho foi apresentado um estudo de uma variante do Problema de Roteamento de Veículo Periódico, denominado Problema do Roteamento Periódico de Unidades Móveis de Pistoneio - PRP-UMP. Para a solução aproximada do PRP-UMP foram propostas duas heurísticas de construção, duas de busca local, além de um procedimento de busca intensiva conhecido na literatura como Reconexão de Caminhos (RC).

Para o desenvolvimento dos algoritmos de construção, foram empregados conceitos clássicos amplamente utilizados em heurísticas para solução de problemas de roteamento, como inserção mais próxima e estrutura de pétalas. Em relação às duas buscas locais propostas neste



(a) Alvo Fácil



(b) Alvo Difícil

Figura 2 – Análise Probabilística para a instância 6

trabalho, a primeira tem como finalidade efetuar um refinamento nas rotas diárias já encontradas para as unidades móveis de pistoneio, enquanto a segunda (BLP) explora uma vizinhança mais ampla envolvendo todo período de planejamento. Para avaliar cada versão GRASP e cada módulo de construção e busca local, foram efetuados basicamente três tipos de experimentos.

A primeira bateria de testes foi realizada para comparar soluções heurísticas com soluções exatas em instâncias de pequenas dimensões. Neste caso, foi utilizada a formulação matemática do PRP-UMP descrevendo-o como um Problema de Programação Linear. Esta formulação foi utilizada para obter soluções exatas para instâncias de pequeno porte de modo a possibilitar a comparação entre os limites obtidos pelas heurísticas com o valor ótimo. Nestes testes pode-se verificar que as heurísticas sempre atingiram o valor ótimo obtido pelo método exato (CPLEX), mas com tempo bem inferior aos exigidos pela resolução exata.

Na segunda bateria de testes, o objetivo foi comparar o desempenho dos algoritmos GRASP propostos neste trabalho. Na avaliação dos GRASP, observou-se que as versões que usam a construção baseada na inserção mais próxima obtiveram desempenho médio superior às versões que utilizaram o construtivo baseado na estrutura de pétalas. Em termos de busca local, foi observado nos testes empíricos que a ordem em que as duas buscas são utilizadas pode alterar o resultado final do GRASP. Em geral, quando a busca local diária foi utilizada antes da busca local periódica implicou em melhoria nas soluções, não mantendo o mesmo comportamento quando aplicada após a busca local periódica. Uma explicação para este fato seria que a busca local periódica pode realizar uma busca mais ampla, e os movimentos realizados que afetam todos os dias do período tendem também a refinar alocações diárias do problema.

Após a análise de desempenho das versões GRASP puras, foi verificado o impacto da introdução do módulo de Reconexão de Caminhos (RC) nestes algoritmos, criando assim as versões GRASP+RC. Os resultados obtidos foram muito promissores. Comparando cada GRASP básico com sua versão GRASP+RC esta última sempre produziu soluções de melhor qualidade em tempos computacionais pouco maiores.

A Análise Probabilística Empírica, terceira e última etapa de testes, mostrou novamente a superioridade de algumas versões incluindo Reconexão de Caminhos, dado que estas convergiram um maior número de vezes para o alvo estipulado. O bom desempenho das versões que utilizam RC mostra que o uso de algum tipo de memória (no caso, armazenar um conjunto elite) tende a produzir bons resultados mostrando um caminho promissor para versões adaptativas do GRASP.

Como trabalhos futuros, pode-se incluir o uso de outras formas de busca local ou o uso de diferentes estratégias de busca numa vizinhança como, por exemplo, o VNS. Devido ao seu bom comportamento na solução aproximada de diferentes problemas de roteamento, a metaheurística Busca Tabu poderia ser verificada para resolução do PRP-UMP.

Agradecimentos

Os autores agradecem a CAPES e CNPq pelo financiamento parcial deste trabalho.

Bibliografia

Aiex, R., Resende, M. G. C., e Ribeiro, C. C. (2002) **Probability distribution of solution time in GRASP: An experimental investigations**. Journal of Heuristics, 8:343–373.

Aloise, D. J., Aloise, D., Rocha, C. T. M., Ribeiro Filho, J. C., Moura, L.S.S. e Ribeiro, C. C. (2004) **Scheduling Workover Rigs for Onshore Oil Production**

Bastos, L. O., Ochi, L. S., e Macambira, E.M. (2005) **GRASP with Path Relinking for the SONET Ring Assignment Problem**. Proc. of the 5th International Conference on Hybrid Intelligent Systems (HIS2005), pp. 239-244, co-sponsored by IEEE Systems, Man, and Cybernetics Society.

Christofides, N. e Beasley, J. E. (1984). **The period routing problem**. Networks, 14:237– 256.

Feo, T. A. e Resende, M. G. C. (1995). **Greedy randomized adaptive search procedures**. Journal of Global Optimization, (6):109–133.

Glover, F. (1996). **Tabu search and adaptive memory programming - advance, applications and challenges**. pages 1–75. R. S. Barr, R. V. Helgason and J. L. Kennington.

Gonçalves, L. B., Martins, S. L. e Ochi, L. S. (2005) **A GRASP with Adaptive Memory for a Period Vehicle Routing Problem**. Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation - CIMCA2005 - M. Mohammadian (editor), volume I, pp.721-727.

Pinto, J. W. C., Martins, S. L. e Rosseti, I. (2006) **Heurísticas para a Resolução do Problema de Roteamento com Número Limitado de Caminhos**. Anais do XXXVIII Simpósio Brasileiro de Pesquisa Operacional, 1357-1368.

Silva, G. C., Andrade, M. R. Q., Ochi, L. S., Martins, S. L. e Plastino, A. (2007) **New Heuristics for the Maximum Diversity Problem**. Status: To appear in Journal of Heuristics - SPRINGER.