Anais do IX Congresso Brasileiro de Redes Neurais / Inteligência Computacional (IX CBRN)
Ouro Preto 25-28 de Outubro de 2009

UM ALGORITMO EVOLUTIVO HÍBRIDO PARA O PROBLEMA DE RECOBRIMENTO DE ROTAS COM COLETA DE PRÊMIOS

MATHEUS S. A. SILVA, MARCIO T. MINE, LUIZ S. OCHI,

Instituto de Computação - Universidade Federal Fluminense
Rua Passo da Pátria 156 - Bloco E - 3º andar - São Domingos. CEP.: 24210-240
Niterói, RJ, Brasil

MARCONE J. F. SOUZA

Departamento de Computação - Universidade Federal de Ouro Preto
Campus Universitário, Morro do Cruzeiro. CEP.: 35400-000
Ouro Preto, MG, Brasil
E-mails: {msalves;mmine;satoru}@ic.uff.br, marcone@iceb.ufop.br

Abstract— This paper proposes a hybrid evolutionary algorithm for getting an approximate solution for the Prize Collecting Covering Tour Problem (PCCTP). The proposed algorithm combines heuristic strategies based on Iterated Local Search, Variable Neighborhood Descent, Path Relinking and GENIUS procedures. Computational results on a set of instances illustrate the effectiveness and the robustness of the proposed heuristic.

Keywords— Computational Intelligence, Metaheuristics, Evolutionary Algorithm.

Resumo— Este artigo propõe um algoritmo evolutivo híbrido para obter soluções aproximadas para o Problema de Recobrimento de Rotas com Coleta de Prêmios (PRRCP). O algoritmo proposto combina estratégias heurísticas baseadas nos procedimentos Busca Local Iterada, Busca em Vizinhança Variável, Reconexão por Caminhos e GENIUS. Resultados computacionais para um conjunto de instâncias mostram a eficiência e a robustez da heurística proposta.

Palavras-chave— Inteligência Computacional, Metaheurísticas, Algoritmo Evolutivo.

1 Introdução

Com o aumento da complexidade dos processos produtivos, há uma necessidade cada vez maior da utilização de sistemas inteligentes, que auxiliem o processo de tomada de decisão da melhor maneira possível. Métodos clássicos de otimização têm encontrado dificuldade para obter a melhor solução, dita ótima, mesmo quando alguns deles possuem teoricamente a garantia de atingí-la. A elevada complexidade dos problemas de otimização encontrados em diferentes áreas tem provocado a necessidade de desenvolvimento de novos métodos mais eficientes na prática para solucionar tais problemas. Esses métodos são usualmente o resultado da adaptação de conceitos de várias áreas. Um exemplo bem sucedido são as metaheurísticas, ou heurísticas inteligentes. A principal característica desta categoria de métodos é a possibilidade de encontrar diferentes ótimos locais durante a busca pela melhor solução. Entre esses métodos, destacam-se os Algoritmos Evolutivos, os quais têm se mostrado eficientes na resolução de vários problemas combinatórios.

Neste trabalho, desenvolvemos um Algoritmo Evolutivo Híbrido (AEH) para resolver o Problema de Recobrimento de Rotas com Coleta de Prêmios (PRRCP). O AEH combina estratégias heurísticas baseadas nos procedimentos *Iterated Local Search*, Busca em Vizinhança Variável (Mladenovic e Hansen, 1997), GENIUS (Gendreau *et al.*, 1992) e Reconexão por Caminhos (Glover e Kochenberger, 2003).

O PRRCP, referido na literatura como *Prize Collecting Covering Tour Problem* (PCCTP), é uma variante do Problema de Recobrimento de Rotas (PRR), que por sua vez é uma generalização do Problema do Caixeiro Viajante (PCV).

O PRR pode ser definido em um grafo não-direcionado $G = (N, E)$, sendo $N = V \cup W$, com $V = T \cup (V \setminus T)$ representando o conjunto dos vértices que podem fazer parte da rota (solução), W o conjunto dos vértices que precisam ser cobertos, $E = \{(v_i, v_j) \mid v_i, v_j \in V\}$, T o conjunto dos vértices obrigatórios, isto é, que *devem* fazer parte da rota e $V \setminus T$, o conjunto dos vértices *opcionais*, que não necessariamente precisam fazer parte da rota. Diz-se que um vértice $w_j \in W$ está coberto quando existe na solução pelo menos um vértice $v_i \in V$ tal que $d(w_j, v_i) \leq D$, sendo D um parâmetro do problema e d uma função real que retorna a distância entre os vértices w_j e v_i.

A partir destas considerações, tem-se que o PRR consiste em determinar uma rota de comprimento mínimo sobre um subconjunto de V, contendo todos os vértices obrigatórios T e cobrindo todos os vértices de W. Uma vez que o PRR pode ser reduzido ao PCV fazendo-se $D = 0$, $W = \varnothing$ e $N = T$, e este se enquadra na classe de problemas *NP-difíceis*, então o PRR também o é.

O PRRCP, além das restrições comuns ao PRR, possui as seguintes particularidades: (*i*) a cada vértice *i* de *V* está associado um prêmio não-negativo p_i; (*ii*) a quantidade dos prêmios coletados nos vértices presentes na solução tem que ser maior ou igual à quantidade mínima pré-estabelecida, dada por *PRIZE*. O objetivo do PRRCP, assim como no PRR, é encontrar uma rota de comprimento mínimo em um subconjunto de *V*, satisfazendo as restrições de pertinência de todos os vértices de *T* à solução, a cobertura de todos os vértices de *W* e por fim, a coleta da quantidade mínima de prêmios (*PRIZE*). Analogamente ao PRR, o PRRCP também é considerado um problema *NP-Difícil*, uma vez que pode ser reduzido ao PCV ($D = 0$, $N = T$, $PRIZE = 0$ e $W = \emptyset$).

2 Trabalhos Relacionados

O PRR foi formulado inicialmente por Current e Schilling (1989). Nesse trabalho, os autores abordam o PRR através de uma heurística de duas etapas, sendo que a primeira minimiza o comprimento da rota e a segunda maximiza o número de vértices cobertos pela rota gerada na primeira etapa. Posteriormente, (Gendreau *et al.*, 1995) propuseram um procedimento baseado na heurística GENIUS (Gendreau *et al.*, 1992) e no Algoritmo PRIMAL *Set Covering* (Balas and Ho, 1980) para resolução do PRR.

Além desse procedimento, nesse trabalho também foram propostas quatro regras de redução, um algoritmo exato baseado na técnica *Branch-and-Cut* e uma formulação matemática. Maniezzo *et al.* (2005) apresentaram uma formulação matemática para o PRR. Além disso, os autores também apresentam heurísticas *Scatter Search* para o PRR. Em Motta *et al.* (2001) é proposta uma nova formulação matemática e uma heurística GRASP para resolver o PRR.

Por ser um problema relativamente novo, existem poucos trabalhos na literatura relacionados ao PRRCP. Em Lyra (2004) são propostas uma formulação matemática, uma regra de redução dos vértices do problema e uma heurística GRASP (Feo e Resende, 1995). Na dissertação de Silva (2009) são propostas três novas regras de redução para o PRRCP, uma nova formulação matemática e duas abordagens heurísticas baseadas em *Iterated Local Search*.

3 Metodologia proposta

Para gerar uma solução inicial são utilizados cinco algoritmos construtivos: 1) Os algoritmos ADD e DROP de Lyra (2004); 2) Adaptação ao PRRCP dos algoritmos de Inserção Mais Barata e do Vizinho Mais Próximo e 3) Adaptação da heurística GENIUS (Gendreau *et al.*, 1992).

Para explorar o espaço de soluções, utilizam-se cinco movimentos clássicos usados para o PCV, os quais são apresentados a seguir: (a) *Shift* - consiste em remover um vértice e reinserí-lo em outra posição; (b) *Swap* - troca dois vértices de suas posições; (c) *Or-Opt* - extensão do *shift*, consistindo em remover um conjunto de vértices adjacentes e os reinserí-los em outra posição; (d) *2-Opt* - remove duas arestas não adjacentes e insere duas novas arestas (e) *3-Opt* - remove três arestas não adjacentes e insere três novas arestas. Além desses movimentos, são usados quatro movimentos baseados nas fases de inserção e remoção da heurística GENIUS e na inserção do algoritmo de Inserção Mais Barata. Para avaliar uma solução, utiliza-se uma função $f(s)$, baseada em penalidade, que determina os custos totais de deslocamento. Essa função consiste em somar todos os custos (distâncias) de todas as arestas presentes na solução e penalizar o não atendimento das restrições do problema.

4 Algoritmo Evolutivo Híbrido (AEH) proposto

O termo Algoritmo Evolutivo (AE) compreende uma família de resolvedores de problemas com base em princípios que podem ser encontrados na evolução biológica (Eiben and Rudolph, 1999).

As três linhas principais de estudo nessa área são o *Algoritmo Genético* (AG), *Estratégias de Evolução* (EE) e *Programação Evolutiva* (PE).

Um AE típico funciona da seguinte forma: a cada geração constrói-se um conjunto de indivíduos, representando as soluções de um problema de otimização. Esses indivíduos, juntamente com a população vinda de gerações anteriores, são combinados numa fase de cooperação para formarem novos indivíduos. Esses novos indivíduos passam, então, por uma fase de adaptação antes de se decidir quais serão incluídos na população que irá para a próxima geração. O algoritmo finaliza ao se atingir um número máximo de gerações (Hertz and Kobler, 2000).

No Algoritmo Evolutivo proposto para resolver o PRRCP, utiliza-se o conceito de classes para diferenciar os indivíduos. A classe *A*, representada pelo *pool* de soluções elite, contém os melhores indivíduos em termos de custo da solução, encontrados durante as gerações. Esses indivíduos são substituídos somente se melhores indivíduos forem criados. A classe *B* é representada por uma parcela da população que também só é substituída caso indivíduos de melhor custo sejam gerados.

Por fim, a classe *C* é representada pela outra parcela restante da população com indivíduos de custos mais elevados, substituídos em todas as gerações. Para gerar a população inicial, utilizam-se versões randomizadas dos cinco métodos construtivos apresentados na Seção 3.

Como os métodos possuem idéias diferentes, diversificando-os na geração da população inicial, diversificam-se os indivíduos, promovendo assim uma miscigenação entre os indivíduos.

O Algoritmo 1 apresenta o pseudocódigo do Algoritmo Evolutivo Híbrido proposto.

```
procedimento Algoritmo Evolutivo Híbrido
   enquanto (!completo (populacao, pool) )
      s0 ← SolucaoInicial
      s ← MRD(s0)
      pool ← s
      se !poolAtualizado então
         populacao ← s
      fim se
   fim enquanto
   enquanto (ger < germax)
      sbase ← s ∈ populacao
      sguia ← s ∈ pool
      sRC ← Reconexão por Caminhos(sbase,sguia)
      s ← ILS-VNRD(sRC)
      pool ← s
      para (nind < classeC)
         s0 ← SolucaoInicial
         s ← MRD(s0)
         populacao ← s
      fim para
      ger ← ger + 1;
   fim enquanto
   s ← s* ∈ pool
retorne s
```

Algoritmo 1: Algoritmo Evolutivo Híbrido

Neste algoritmo, inicialmente são criados vários indivíduos para compor uma *populacao* e o *pool* de soluções elite. Cada indivíduo é então submetido a uma busca local realizada pelo Método Randômico de Descida - MRD (descrito na Subseção 4.1). Se esse indivíduo *s* for melhor que o indivíduo que possuir o pior custo dentre todas as soluções do *pool* e possuir um percentual mínimo de diferença para todas as soluções desse conjunto elite, então se atualiza o *pool* com a inclusão de *s*. Caso esse indivíduo não satisfaça os requisitos para entrar no *pool* (*poolAtualizado*), verifica-se se *s* pode fazer parte da *populacao*. Para isso é preciso que ele seja melhor que a pior solução desse conjunto, não precisando ter um percentual mínimo de diferença em relação aos demais indivíduos. Criada a *populacao* inicial, enquanto o número máximo de geração (ger_{max}) não for alcançado, aleatoriamente escolhe-se uma solução da *populacao* para ser a solução base do procedimento *Reconexão por Caminhos* (vide Subseção 4.2). A solução guia é selecionada do *pool* de soluções elite de acordo com o grau de diferença em relação à solução base. A solução que possuir o maior grau de diferença será escolhida como guia. A melhor solução encontrada durante a *Reconexão por Caminhos* é então submetida a uma busca local realizada pelo procedimento *ILS-VNRD* descrito na Subseção 4.3. Caso a solução *s* retornada pelo *ILS-VNRD* seja melhor que a solução de maior custo do *pool* e se *s* possuir uma porcentagem mínima de diferença para as demais soluções do conjunto elite, atualiza-se o *pool* com *s*. Ao final de cada geração reconstrói-se a classe *C* da *populacao*, substituindo-se seus indivíduos por novos indivíduos. Como a *populacao* é ordenada pelo custo dos indivíduos, pode acontecer de um novo indivíduo criado para a classe *C* possuir um custo menor que um indivíduo da classe *B*. Quando isto ocorre, realiza-se a inversão de classes entre os indivíduos, isto é, automaticamente o novo indivíduo passa a ser da classe *B* e o outro indivíduo que pertencia a esta classe passa agora a pertencer à classe *C*. Ao final do algoritmo, retorna-se o melhor indivíduo (solução) do *pool*.

4.1 Método Randômico de Descida – MRD

O Método Randômico de Descida, ou MRD, é uma heurística de refinamento que consiste em analisar um vizinho qualquer de uma dada solução e o aceitar somente se ele for estritamente melhor que a solução corrente. Caso esse vizinho não seja melhor, a solução corrente permanece inalterada e outro vizinho é gerado. O procedimento é finalizado quando se atinge um número máximo de iterações sem que haja melhorias no valor da melhor solução obtida.

4.2 Reconexão por Caminhos – RC

O procedimento de Reconexão de Caminhos (Glover e Kochenberger, 2003) foi utilizado como fase de intensificação para o AEH proposto. Este algoritmo inicia calculando a diferença simétrica de todas as soluções do *pool* de soluções elite em relação à solução base, s_{base}, proveniente da *populacao*. A diferença simétrica pode ser entendida como a quantidade de passos que são necessários para sair de uma solução base e chegar a uma solução guia. Isto significa que a cada iteração, um atributo que exista na solução guia, mas que não pertença à solução base, é inserido nesta com o objetivo de que ao final do procedimento a solução base coincida com a solução guia.

Definida a solução base, s_{base}, enquanto ela possuir alguma diferença para s_{guia}, o procedimento aplica um movimento que consiste em inserir na s_{base} um atributo de s_{guia}. O movimento escolhido é o melhor dentre os possíveis movimentos que podem ser aplicados, isto é, o que trouxer o maior benefício para a solução. A essa solução intermediária, também chamada de candidata (s_{cand}), aplica-se uma busca local por meio do MRD. Se essa solução *s* for melhor que a pior solução do *pool* e atender ao critério de diversificação de soluções, então se atualiza o *pool* com *s*.

O procedimento termina quando ocorre a convergência das soluções, isto é, quando as soluções base e guia possuem os mesmos vértices e a ordem de visita dos vértices em ambas as soluções é a mesma.

4.3 ILS-VNRD

O procedimento ILS-VNRD tem como solução inicial a melhor solução encontrada durante a Reconexão por Caminhos (Subseção 4.2). Em seguida, aplica-se um mecanismo de busca local, o VNRD (Subseção 4.4). A cada iteração do procedimento ILS-VNRD, gera-se uma perturbação na solução corrente. Essa perturbação consiste em realizar *k* movimentos Aleatórios dentre aqueles descritos na Seção 3, sendo *k* um número arbitrário compreendido entre *1* e um valor parametrizado, denominado *kp*.

A essa solução perturbada, aplica-se o VNRD, tentando com isso gerar uma solução melhorada. Se essa for melhor que a solução corrente, então a solução melhorada é aceita como a nova solução corrente e reinicia-se o valor de *kp*. Essa repetição é executada até que um número máximo de iterações sem melhora seja realizado. Quando isso ocorre, incrementa-se o grau de perturbação, *kp*, em um fator *delta*, até que *kp* atinja seu valor máximo (kp_{max}). A variação do grau de perturbação é uma estratégia importante, pois permite a intensificação da busca e diversificação das soluções geradas. A busca é intensificada quando o grau de perturbação é baixo e a diversificação ocorre quando esse grau atinge um valor relativamente alto.

4.4 Descida Randômica em Vizinhança Variável

O Método de Descida Randômica em Vizinhança Variável (*Variable Neighboohoord Randon Descent*, VNRD), é uma variante do Método de Descida em Vizinhança Variável (*Variable Neighboohoord Descent*, VND), proposto por Nenad Mladenovic e Pierre Hansen em 1997. O VND é um método de refinamento que consiste em explorar o espaço de soluções por meio de trocas sistemáticas de estruturas de vizinhança (descritas na Seção 3), aceitando somente soluções de melhora da solução corrente e retornando à primeira estrutura quando uma solução melhor é encontrada. A grande diferença do VNRD para o VND é que a exploração de cada vizinhança não é feita por completa, apenas um percentual desta é explorada. O valor desse percentual é um parâmetro passado para o método. Este método finaliza quando se aplica todas as estruturas de vizinhança e não se encontra nenhuma solução de melhora.

5 Resultados e Análise

O AEH foi implementado na linguagem C++ utilizando o ambiente Microsoft Visual Studio 2008 e testado em um computador Intel Core 2 Duo, com 2.2 GHz e 2.5 GB de memória principal, rodando o sistema operacional Windows Vista. Para validá-lo, foram utilizados problemas-teste adaptados para o PRRCP a partir do TSPLIB, proposto em Silva (2009). A justificativa para não se ter utilizado os problemas-teste do trabalho de Lyra (2009) se deve à não disponibilidade de tais problemas na literatura. Em Silva (2009) foram criados problemas-teste com coleta mínima de 25%, 50% e 75% do prêmio total dos vértices de *V*. Com isso, em grande parte dos problemas, alguns vértices do conjunto *V\T* devem estar obrigatoriamente presentes na solução, o que dificulta a resolução do problema, pois o número de combinações aumenta exponencialmente.

Foram utilizados dois grupos de problemas para validar a metodologia proposta. Esses grupos, chamados de Grupo 1 e Grupo 2 respectivamente, possuem problemas com o número de vértices variando entre *50* e *200* para o Grupo 1 e de *201* a *400* para o Grupo 2. Para cada problema foram realizadas *10* execuções, cada qual partindo de uma semente diferente de números aleatórios. Os parâmetros adotados no Algoritmo Evolutivo foram: *maxGer* = 7, *populationSize* = 5, *classBSize* = 3, *classCSize* = 2, *eliteSetSize* = 5, *percDiffSolution* = 0,15, *iterMaxMRD* = 300, *iterMaxILS* = 100, *iterMaxVNRD* = 100, *kpMinILSVNRD* = 5, *kpMaxILSVNRD* = 7 e *delta* = 2. Tais valores foram definidos através de testes preliminares.

Para legitimar o desempenho do AEH proposto, este foi comparado com as duas melhores versões, descritas em Silva (2009), de um algoritmo de busca local baseado na metaheurística *Iterated Local Search*, tendo como método de busca local o MRD. Para o Grupo 1, o AEH foi comparado com a versão ILS-MRD_AD, a qual tem o procedimento ADD como método de geração de solução inicial. Para o Grupo 2, o AEH foi comparado com a versão ILS-MRD_IB, a qual usa o procedimento da Inserção Mais Barata para construir uma solução.

Esses algoritmos foram comparados pelos seguintes critérios: (a) *custo médio de todos os problemas do grupo;* **(b)** *desvio médio das soluções em relação ao melhor custo conhecido*; **(c)** *taxa de sucesso de cada algoritmo em alcançar o melhor custo em pelo menos uma das dez execuções;* **(d)** *porcentagem de problemas que cada algoritmo encontrou o menor custo sozinho*

De acordo com os critérios considerados, observou-se que, para os problemas do Grupo 1, o AEH obteve um desempenho superior ao ILS-MRD_AD, já que o custo médio das soluções obtidas pelo AE foi de *20073* enquanto o ILS-MRD_AD obteve um custo médio de *20080*. O desvio médio das soluções do AE foi inferior ao do ILS-MRD_AD, no caso, de *0,02*% contra *0,06*% do outro algoritmo. Além disso, o AEH encontrou o melhor valor conhecido em 99,1% dos casos do Grupo 1 contra os 96,4% obtidos pelo ILS-MRD_AD, e em 1,8% dos problemas-teste, o AE encontrou esse melhor valor sozinho enquanto o outro algoritmo não obteve sucesso nesse critério.

Em relação aos problemas do Grupo 2, observa-se que a diferença do desempenho do AEH em relação ao ILS-MRD_IB, avaliado pelos critérios propostos, foi mais significativa. De fato, o AEH obteve um custo médio das soluções de *33641*, enquanto o ILS-MRD_IB obteve um custo médio de *33752*. O desvio médio das soluções do AE também foi inferior ao do ILS-MRD_IB, no caso, de *0,14*% contra *0,34*% do outro algoritmo. Além disso, o AEH encontrou o melhor valor conhecido em mais de *90*% dos casos do Grupo 1 contra os *76*,7% obtidos pelo ILS-MRD_IB, e em *16,7*% dos problemas-teste, o AEH superou a melhor solução conhecida nos problemas-teste desse grupo. Ainda em relação as instâncias do Grupo 1, foi mostrado em Silva (2009) a solução ótima de parte dos problemas-teste deste grupo utilizando uma formulação matemática e o *software* C-PLEX 11; e tanto o ILS-MRD_IB como o AEH alcançaram o valor ótimo sempre que este é conhecido.

Testes computacionais foram efetuados usando como critério de parada um valor alvo a ser atingido. Foi verificado que o AEH possui uma convergência empírica melhor quando se deseja encontrar uma solução de boa qualidade, ou seja, se colocarmos um valor alvo *difícil* (solução ótima ou sub-ótima) o AEH tende a ser mais rápido do que o algoritmo ILS-MRD_IB. Estes testes mostram que apesar do AEH possuir um tempo computacional maior, justificado pelo número de operações de otimização que realiza a cada iteração, este tende a requerer na média menos iterações para obter um valor alvo de boa qualidade.

De acordo com os resultados apresentados, o AEH foi o que gerou as melhores soluções para a maioria dos problemas-teste. Esse mérito se deve à característica do algoritmo em trabalhar com os dois tipos de buscas, local e populacional, mesclando soluções de boa qualidade com soluções de qualidade inferior, fazendo com que a busca local tenha mais chances de melhorar a solução. O algoritmo se mostrou robusto por apresentar soluções finais com baixa variabilidade, no caso, sempre inferior a *0,14*%.

5 Conclusões

Este trabalho abordou o Problema de Recobrimento de Rotas com Coleta de Prêmios (PRRCP). Para resolvê-lo, foi proposto um Algoritmo Evolutivo Híbrido, que mescla busca local com busca populacional. Para gerar os indivíduos da população foram adaptados cinco métodos de geração de solução inicial do PCV para serem utilizados no PRRCP. Para combinar diferentes indivíduos, foi utilizado o procedimento Reconexão por Caminhos e para refinar esses indivíduos, utilizou-se a metaheurística *Iterated Local Search* (ILS), tendo o *Variable Neighborhood Random Descent* (VNRD) como método de busca local. O VNRD explora a vizinhança de uma solução por meio de várias estratégias apresentadas na Seção 3. De acordo com os resultados obtidos, verifica-se que o algoritmo proposto é eficiente e robusto para resolver o PRRCP, sendo capaz de produzir soluções de boa qualidade.

Referências Bibliográficas

Balas, E. and Ho, A. (1980). Set covering algorithms using cutting planes, heuristics, and subgradient optimizations: A computational study. *Mathematical Programming*, 12:37-70.

Current, J. R. and Schilling, D. A. (1989). The covering salesman problem, *Transportation Science*. 23: 208-213.

Eiben, A. E. and Rudolph, G. (1999). Theory of evolutionary algorithms: A birds eye view. *Theoretical Computer Science*, 229: 3-9.

Feo, T. and Resende, M. (1995). Greedy randomized search procedure, *Journal of Global Optimization*, 6: 109-133.

Gendreau, M., Hertz, A. and Laporte, G. (1992). New insertion and post optimization procedures for the traveling salesman problem, *Oper. Res.*, 40: 1086-1094.

Gendreau, M., Laporte, G. and Semet, F. (1995). The Covering tour problem, *Oper. Res.*, 45: 568-576.

Glover, F.; Kochenberger, G. *Handbook of Metaheuristics*. Kluwer, 2003.

Hertz, A. and Kobler, D. (2000). A framework for the description of evolutionary algorithms. *European Journal of Oper. Res.*, 126:1-12.

Lyra, A. R. de. (2004). O problema de recobrimento de rotas com coleta de prêmios: Regras de redução, formulação matemática e heurísticas, Master's thesis, IC/UFF.

Maniezzo, V., Baldacci, R. and Zamboni, M. (2005). Scatter search methods for the covering tour problem, In Metaheuristic Optimization via Memory and Evolution: Tabu Search and Scatter Search, vol 30: 59-91. Kluwer.

Mladenovic, N. and Hansen, P. (1997). Variable neighboorhood search, *Computers and Operations Research*, 24: 1097-1100.

Motta, L. C. S., Ochi, L. S. and Martinhon, C. A. (2001). Reduction rules for the covering tour problems, *Eletronic Notes In Discrete Applied Mathematics*, 7: 168-171.

Silva, M. de S. A. (2009). Problema de Recobrimento de Rotas com Coleta de Prêmios, Master's thesis, IC/UFF.