Publicado nos Anais (em CD-ROM) do XXXII Simpósio Brasileiro de Pesquisa Operacional (XXXII SBPO), 10/2000
páginas: 137-150, ISBN: 1518-1731

O Problema de Recobrimento de Rotas: Formulação, Testes de Redução e Soluções Aproximadas via GRASP

Luciene Cristina Soares Motta, Luiz Satoru Ochi e Carlos Alberto Martinhon

Universidade Federal Fluminense
Programa de Pós Graduação em Ciência da Computação - Instituto de Computação
Correspondência: Rua Capitão Zeferino 56/808 - Icaraí - Niterói - RJ - Brasil - 24220-230
E-mail: luciene.motta@pegasus.pgcc.uff.br; satoru@dcc.ic.uff.br

palavras-chave: otimização combinatória, metaheuristicas, algoritmos adaptativos.

Abstract

The Covering Tour Problem (CTP) is a job sequencing problem and it is defined on a graph $G = (V \cup W, E)$, where W is a set of vertices that must be covered. The problem consists of determining a minimum length Hamiltonian cycle on a subset of V such that every vertex of W is within a given distance d from at least one node in the cycle. Being a generalization of the Traveling Salesman Problem (TSP), the CTP is NP-Hard. This work presents a new mathematical formulation based on flow variables, reduction rules for the associated graph and two original metaheuristc algorithms to solve the CTP approximately.

Resumo

O Problema de Recobrimento de Rotas (PRR) é um problema de sequenciamento de tarefas definido sob um grafo $G = (V \cup W, E)$, onde W é o conjunto de vértices que devem ser cobertos. O problema consiste em determinar uma rota ou um ciclo Hamiltoneano de comprimento mínimo sob um subconjunto de V de modo que todo vértice de W diste no máximo d de algum vértice da rota. Sendo uma generalização do Problema do Caixeiro Viajante (PCV), o PRR é considerado NP-difícil. Este trabalho apresenta uma nova formulação matemática baseada em variáveis de fluxo, regras para a redução do grafo associado e duas metaheuristicas originais para solucionar aproximadamente o PRR.

1. Introdução

O impacto das novas tecnologias em computação tem apresentado uma necessidade cada vez maior de resolver problemas até então desconsiderados. Diante das diversas mudanças na resolução de problemas difíceis de otimização, que englobam os problemas do mundo real, os métodos clássicos de otimização têm encontrado grandes dificuldades na obtenção da solução ótima (ótimo global), mesmo quando estes métodos possuem teoricamente a garantia de atingi-la.

A grande quantidade de problemas de otimização de elevada complexidade encontrados em diferentes áreas, tem provocado a necessidade de se pesquisar cada vez mais o desenvolvimento de novos métodos eficientes não só do ponto de vista teórico, como também do ponto de vista operacional. Esses métodos são usualmente o resultado da adaptação de idéias de uma grande variedade de áreas, esperando que se produza procedimentos eficientes e hábeis para lidar com problemas cada vez mais complexos e dinâmicos. Algumas dessas idéias tem motivado o surgimento de estruturas com novas metodologias, como por exemplo a dos Algoritmos Genéticos, que procuram imitar o fenômeno biológico da reprodução evolutiva; as Colônias de Formigas, que simulam o comportamento das formigas que cooperam entre si; as Redes Neurais e até outros baseados em fenômenos físicos como o *Simulated Annealing*. Estes procedimentos são chamados de metaheuristicas e possuem como principais características a existência de ferramentas que possibilitam se esquivar de ótimos locais ainda distantes da solução ótima do problema (ótimo global) e a flexibilidade, responsável por uma melhor adaptação ao espaço de busca.

O objetivo deste trabalho é propor: uma nova modelagem matemática para o Problema de Recobrimento de Rotas, um estudo de regras para a redução do grafo de instância, assim como novos algoritmos aproximados: um para a construção de uma solução viável e o outro para a busca local. Afim de que sejam mais eficientes que os convencionais, compartilhamos características de duas metaheuristicas que tem se destacado na solução de problemas de otimização combinatória pelos resultados obtidos: GRASP e VNS.

Propostos para solucionar aproximadamente o Problema de Recobrimento de Rotas (PRR), os algoritmos apresentados podem ser facilmente estendidos para outros problemas cuja característica seja encontrar uma rota sob um subconjunto dos vértices de um grafo. O PRR foi escolhido, pois apesar da sua relevância prática, não tem recebido muita atenção da comunidade afim. Ao que é de nosso conhecimento, somente uma metaheuristica (*Scatter Search*) foi proposta para este problema até o presente momento.

2. O Problema de Recobrimento de Rotas

Inicialmente definido por John Current [1], o Problema de Recobrimento de Rotas (PRR), referido na literatura por *Covering Tour Problem* (CTP), é uma generalização do Problema do Caixeiro Viajante (*Traveling Salesman Problem* - TSP) e possui diversas aplicações práticas incluindo roteamento.

O PRR pode ser definido da seguinte forma: seja $G = (V \cup W, E)$ um grafo completo, não direcionado, onde $V \cup W = \{v_1, ..., v_n\}$ é o conjunto de vértices, e $E = \{(v_i, v_j) \mid v_i, v_j \in V \cup W, i < j\}$ é o conjunto de arestas. O vértice v_1 é o depósito, V é o conjunto dos vértices que podem ser visitados, $T \setminus V$ é o conjunto dos vértices que devem ser visitados $(v_1 \in T)$, e W é o conjunto dos vértices que devem ser cobertos. É suposto que a matriz de distância $C = (c_{ij})$, definida sob o conjunto de arestas E, satisfaz a desigualdade triangular. O problema consiste em determinar uma rota ou um ciclo Hamiltoneano de comprimento mínimo sob um subconjunto de V. Esta rota deve conter todos os vértices de T e deve cobrir cada vértice de W. Um vértice $v_i \in W$ é considerado coberto se distar no máximo d de algum vértice da rota.

O PRR é considerado NP-difícil uma vez que se reduz ao Problema do Caixeiro Viajante quando $d = 0$ e todos os vértices de W coincidem com os vértices de V. Existem na literatura problemas similares ao PRR. Entre eles estão: o *Shortest Covering Path Problem* (SCPP) ([2], [3]) que consiste em determinar um caminho de custo mínimo em uma rede, partindo de um nó origem i e retornando a um nó destino f $(i \neq f)$, ambos previamente estabelecidos. O caminho deve cobrir todos os nós da rede. Um nó é considerado coberto se estiver a uma distância máxima pré-determinada S de um nó do caminho; o *Tour Cover of a Graph* (CTG) [4], onde os custos estão associados aos nós e não \`as arestas; o *Multi-Vehicle Covering Tour Problem* (m-CTP) [5] que é definido sob um grafo $G = (V \cup W, E)$, onde W é o conjunto de vértices que devem ser coletivamente cobertos por no máximo m rotas de custo total mínimo (m é o número de veículos disponíveis); o *Median Tour Problem* (MTP) e o *Maximal Covering Tour Problem* (MCTP) [6], que consistem na minimização do comprimento de uma rota que deve visitar somente p dos n nós da rede e na maximização do acesso dos nós que não estejam nesta rota. A diferença entre os dois problemas é que no MTP deseja-se minimizar o custo total da distância de acesso \`a rota dos nós que não fazem parte dela, enquanto no MCTP deseja-se minimizar a demanda total dos nós não cobertos pelos nós da rota. Podemos também mencionar o *Prize Collecting Traveling Salesman Problem* (PTSP) [7], [8] e o *Selective Traveling Salesman Problem* (STSP) [9] como problemas relacionados ao PRR. Ambos possuem um prêmio p_i, não negativo, associado a cada vértice *i* do grafo e consistem na construção de uma rota através de um subconjunto destes vértices. O PTSP consiste na minimização do custo da rota, onde o total de prêmios coletados é no mínimo *p*. O STSP consiste na maximização do total de prêmios coletados pela rota cujo comprimento total não deve exceder um certo valor *r*. Uma versão contínua do PRR é também chamada *Geometric Covering Salesman Problem* (GCSP) [10], onde o conjunto de vértices não é um conjunto discreto, mas sim a região de um plano.

Existem diversas aplicações para o problema em estudo. Current and Schilling descreveram alguns problemas reais de roteamento que correspondem ao PRR [11]. Entre eles, podemos citar o roteamento de equipes de entrega de medicamentos em países subdesenvolvidos, onde medicamentos somente podem ser entregues a um subconjunto de vilarejos e todos os usuários dos demais vilarejos devem ser capazes de atingir os visitados [12]. Outro problema é a localização de caixas de correios em um subconjunto de locais candidatos [13], onde todos os usuários devem estar localizados a uma distância razoável de alguma das caixas. O custo da rota através de todas as caixas deve ser minimizado.

Podemos ainda citar como aplicações para o PRR o roteamento de aeronaves em vôos noturnos, onde a rota não inclui a visita em todas as cidades diretamente, e o planejamento de paradas de um circo durante uma estação. Este último é também conhecido como *Traveling Circus Problem* [14] e consiste em encontrar uma rota onde as localidades não incluídas no planejamento de paradas estejam acessíveis a pelo menos uma das incluídas. Para cada localidade não visitada acrescenta-se uma penalidade.

3. Literatura Existente

Apesar de apresentar uma grande aplicabilidade a problemas reais, o PRR não tem recebido muita atenção na literatura desde que foi introduzido em 1981. Current and Schilling [11] propuseram uma heurística para gerar um conjunto de soluções eficientes para duas versões distintas do problema. A primeira versão tem como objetivo minimizar o comprimento da rota e a segunda, o número de vértices cobertos por esta. Em 1997, Gendreau, Laporte e Semet [15], analisaram uma nova formulação matemática para o PRR e propuseram um algoritmo exato e uma heurística. A heurística é usada no algoritmo exato para produzir um limite superior inicial, combinando a heurística GENIUS para o TSP [18] com a PRIMAL1 para o *Set Covering Problem* [19]. Um ciclo Hamiltoneano é construído sob os vértices obrigatórios utilizando a heurística GENIUS. Então, caso necessário, vértices de V são inseridos no ciclo de acordo com o seu custo associado, até que todos os vértices de W estejam cobertos. O algoritmo exato é do tipo *Branch-and-Cut* e usa o código CPLEX [20]. Nele, a partir de um nó genérico da árvore, é resolvido um problema linear contendo um subconjunto de restrições válidas. É feita uma busca para restrições violadas e algumas destas restrições são introduzidas no problema corrente que é então reotimizado. Este processo é repetido até que uma solução viável seja obtida ou até que seja mais promissor ramificar a variável fracionária. Os resultados foram aplicados a alguns problemas onde somente um subconjunto de vértices são visitados. Entre eles estão o *Prize Collection Traveling Salesman Problem* (PTSP), o *Selective Traveling Salesman Problem* (STSP) e o *Generalized Traveling Salesman Problem* (GTSP).

Os algoritmos foram testados com uma série de problemas aleatórios. Os $|V|+|W|$ vértices foram gerados aleatoriamente em um quadrado de dimensões [0, 100] x [0, 100] considerando uma distribuição uniforme. Os conjuntos T e V foram definidos considerando os $|T|$ e $|V|$ primeiros pontos respectivamente, e W foi definido como o restante dos pontos. Os coeficientes c_{ij} foram computados como a distância Euclideana entre os pontos e o valor de d foi determinado por:

$$d = \max(\max_{v_k \in V \setminus T} \min_{v_l \in W} \{c_{lk}\}, \max_{v_l \in W} \{c_{l,k(l)}\})$$

onde $k(l)$ é o índice do segundo vértice de $V \setminus T$ mais próximo de v_1. Desta forma, cada vértice de $V \setminus T$ cobre pelo menos um vértice de W e cada vértice de W é coberto por pelo menos dois vértices de $V \setminus T$.

Uma outra regra também foi utilizada para os vértices de W. Estes não foram expressos por coordenadas, mas listados em um vetor e cobertos por cada elemento de V com uma probabilidade *p* tomada sucessivamente como *0.1*, *0.2*, *0.5* e *0.7*. Os algoritmos são codificados em C e foram executados em uma estação SunSPARC 1000. Testes foram realizados com instâncias para $n = 50, 75 \text{ e } 100$, $|T| = 1, \lceil 0.25n \rceil, \lceil 0.50n \rceil, \lceil 0.75n \rceil$ e $|W| = n, 2n, 3n, 4n, 5n.$ Para cada combinação desses parâmetros, 5 instâncias foram resolvidas, envolvendo até 600 nós. As que envolviam até 100 nós, foram resolvidas otimamente com um tempo computacional razoável.

Os testes realizados constataram que a heurística obteve um bom desempenho para instâncias em que $|T|$ era pequeno. Para o algoritmo do tipo *Branch-and-Bound*, o limite inferior foi bem próximo do valor ótimo (cerca de 0.5 por cento). Como conseqüência, a árvore de busca era relativamente pequena, mas um certo esforço computacional era despendido em cada nó.

Em 1999, Maniezzo et al. [25] apresentaram três algoritmos utilizando a metaheuristica *Scatter Search* proposta por Glover [26], [27]. A idéia básica desta metaheuristica é a utilização de um conjunto R de soluções, chamado Conjunto Referência, onde operadores de recombinação são aplicados nos pontos de R, gerando novas soluções que são posteriormente utilizadas como pontos de partida para procedimentos de busca local. As melhores soluções obtidas ao final deste processo são inseridas no Conjunto Referência. Todo o procedimento é repetido por um determinado número de iterações. Geralmente, uma combinação linear é utilizada como operador de recombinação de soluções. O primeiro algoritmo, **SS-CTP1**, utiliza cortes, o segundo, **SS-CTP2**, utiliza ambos, cortes e *star paths* e o terceiro, **SS-CTPB**, não utiliza cortes. Todos os algoritmos compartilham uma série de procedimentos de problemas relacionados ao PRR, por este motivo, um melhor detalhamento pode ser encontrado em [25].

Os algoritmos são codificados em Fortran 77 e foram executados em uma Silicon Graphics Indy (MIPS R4400/processadores de 200Mhz), equipados com 256Mb de RAM em uma Irix 5.3. O CPLEX 4.0 [20] foi utilizado para resolver o problema Lagrangeano. As instâncias foram geradas como propostas por Gendreau et al. [15], como descrito anteriormente. Os custos $\{c_{ij}\}$ foram computados como valores inteiros iguais `a $\lfloor e_{ij} + 0.5 \rfloor$, onde $e_{ij}$ era a distância Euclideana entre os pontos *i* e *j*. Os valores de $D_j$, $j = 1,2,...,n$, eram todos iguais `a *d*, como definido em [15]. Testes comparativos foram realizados com a heurística proposta por Gendreau et al. [15] e foi constatado que esta é muito mais rápida que o *Scatter Search*. Em muitos casos o problema é resolvido otimamente em menos tempo.

Não é de nosso conhecimento nenhum outro algoritmo, aproximado ou exato, proposto para o PRR.

4. Procedimentos Propostos

Uma maneira trivial de se resolver problemas de otimização combinatória é simplesmente listar todas as possíveis soluções e selecionar a melhor. Entretanto, apesar de ser teoricamente viável a utilização desta abordagem, a enumeração completa não funciona na prática. O vasto número de possíveis soluções torna este caminho inviável. Nas últimas décadas, a Pesquisa Operacional concentrou-se tanto na obtenção da solução ótima de problemas de otimização combinatória, como também na modelagem dos problemas da vida real. Consequentemente, vários algoritmos exatos foram desenvolvidos para solucioná-los de modo muito mais eficiente que a enumeração de todas as suas possíveis soluções. Em contrapartida, estes algoritmos não são capazes de encontrar a solução ótima de problemas de larga escala (geralmente problemas reais) em um tempo computacional razoável. Diante desta problemática, as Heurísticas começaram a ser desenvolvidas. O objetivo principal desta técnica é obter uma boa solução, isto é, próxima da solução ótima, em um tempo computacional razoável. Por não possuírem a robustez dos métodos exatos, as heurísticas chegaram a ser consideradas como caminho alternativo para a solução de problemas de elevadas dimensões. Contudo, estas heurísticas convencionais, tem os seus limitantes e o principal deles é a dificuldade de escapar de ótimos locais ainda distantes de um ótimo global. Mais recentemente, a partir de 1985, começaram a surgir heurísticas genéricas conhecidas como Metaheuristicas ou Heurísticas Inteligentes, cuja principal característica é a existência de ferramentas que possibilitam se esquivar de ótimos locais ainda distantes da solução ótima do problema (ótimo global).

Nas seções a seguir, além de enfocar um estudo teórico acerca do PRR, propondo uma descrição mais genérica para o problema, testes para a redução do grafo de instância associado, assim como uma nova formulação de Programação Linear Inteira introduzindo variáveis de fluxo**,** apresentamos também um algoritmo metaheuristico, composto de uma fase de construção e uma de busca local, que compartilha características de duas metaheuristicas conhecidas como GRASP (*Greedy Randomized Adaptive Search Procedures*)[28] e VNS, (*Variable Neigborhood Search*) [16], [17].

4.1. Uma representação para o PRR usando grupamentos

Na Seção 2 apresentamos a descrição original do Problema de Recobrimento de Rotas. Esta descrição foi adotada em todos os trabalhos desenvolvidos acerca deste problema. Com a finalidade de torná-lo mais genérico, adotaremos uma nova descrição. Na forma original, apenas os vértices de V podem fazer parte da solução (isto é, os vértices opcionais $v_i \in V \setminus T$ e os obrigatórios $v_j \in T$). Assumiremos que não só os vértices de V como também os vértices de W (vértices que devem ser cobertos) poderão fazer parte da rota solução do problema. Sendo assim, $v_i \in W$ também poderá ser utilizado para cobrir a si próprio e a outros vértices $v_j \in W, i \neq j$. A *figuras 1* e a *figura 2* ilustram uma solução considerando-se ambas descrições.

Segundo a literatura, o PRR também pode ser formulado como o Problema do Caixeiro Viajante Generalizado (*Generalized Traveling Salesman Problem* - GTSP) [21], o que torna possível definir grupamentos para o problema do seguinte modo:

$$C_i = \{ j \in V \cup W \mid c_{ij} \le d \}, \qquad \forall i \in W \qquad (1)$$

$$C_t = \{t\}, \qquad \forall t \in T \qquad (2)$$

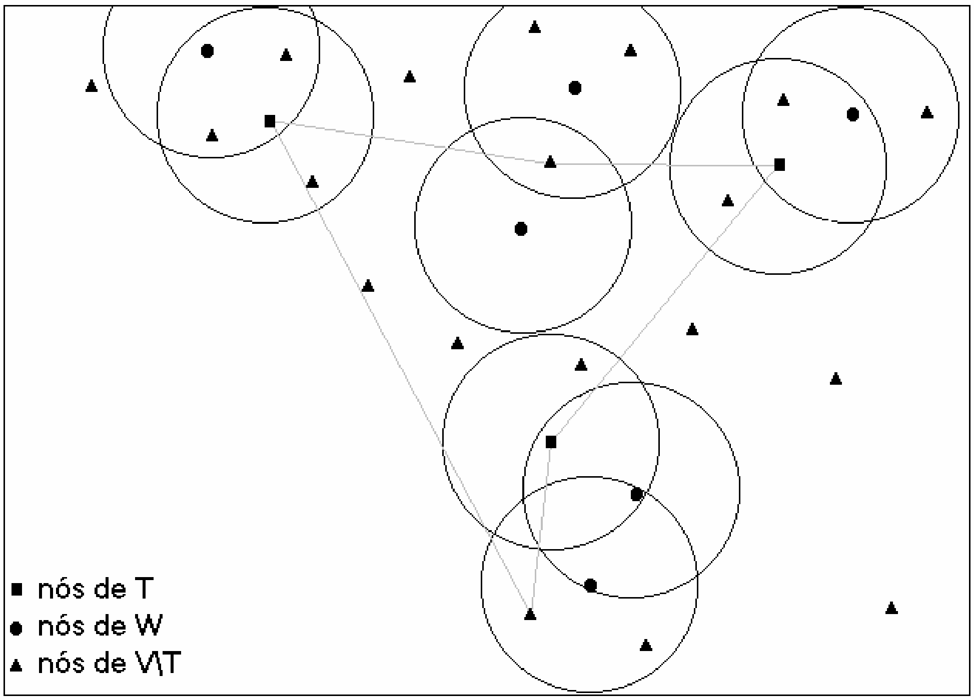

Figura 1: Grafo de instância para o CTP (arestas não estão representadas para melhor visualização). Solução ótima segundo a descrição original apresentada na Seção 2.

4.2. Formulação Proposta

Nesta seção, descreveremos uma nova formulação de Programação Linear Inteira para o PRR introduzindo variáveis de fluxo. Ela apresenta-se interessante em diversos aspectos, entre eles, uma relaxação satisfaz de forma robusta as restrições de eliminação de subciclos, acarretando, do ponto de vista computacional, o crescimento polinomial do conjunto de restrições adicionais com o tamanho do problema, ao contrário do crescimento exponencial das restrições tradicionais do Problema do Caixeiro Viajante.

A idéia básica desta formulação é associar uma variável de fluxo a cada aresta $(v_i, v_j) \in E$.

Seja y_k, para $v_k \in V$, uma variável binária que é igual a 1 se e, somente se, o vértice v_k estiver na rota e zero, caso contrário. Se $v_k \in T$, então y_k é necessariamente igual a 1.

Seja x_{ij}, para todo $v_i, v_j \in V$, com $i \ne j$ uma outra variável binária que é igual a 1 se e, somente se, a aresta (v_i, v_j) pertence `a rota e zero, caso contrário. Também são definidos coeficientes binários δ_{lk} iguais a 1 se e, somente se, $v_l \in W$ puder ser coberto por $v_k \in V$ (isto é, se $c_{lk} \le d$ e $S_l = \{v_k \in V \mid \delta_{lk} = 1\}$ } e zero, caso contrário. A variável de fluxo

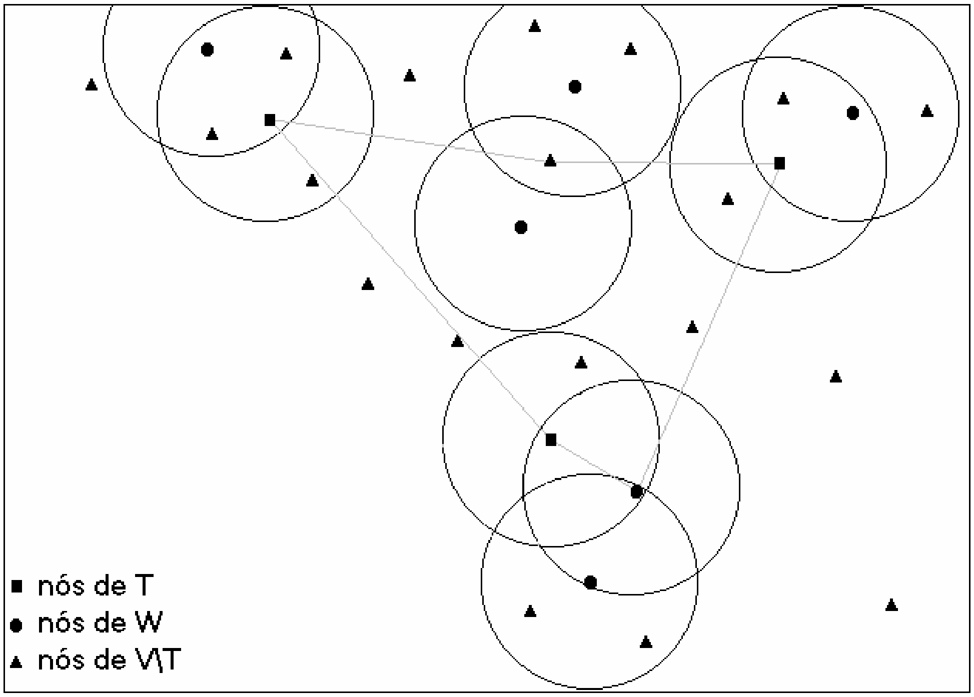

Figura 2: Mesmo grafo de instância para o CTP da *Figura 1*. Solução ótima segundo a descrição proposta na Seção 4.1.

associada a cada $(v_i, v_j) \in E$, com $i \neq j$, é definida como uma variável inteira não negativa, dada por z_{ij}, que representa a quantidade de fluxo escoado no arco (v_i, v_j).

De posse destas considerações e das abordagens adotadas acima para o PRR, a formulação proposta é apresentada da seguinte forma:

$$\text{(PRR) minimizar} \quad \sum_{i \neq j | i,\, j \in (V \cup W)} c_{ij} x_{ij} \ , \qquad (3)$$

sujeito `a:

$$\sum_{v_k \in C_l} y_k \geq 1 \qquad (C_l \text{ definido em (1)}) \qquad (4)$$

$$\sum_{i \neq k} x_{ik} + \sum_{j \neq k} x_{kj} = 2 y_k \qquad (v_k \in V \cup W) \qquad (5)$$

$$\sum_{j \in V \cup W} z_{kj} = \sum_{i \in V \cup W} z_{ik} + y_k \qquad (k \in V \cup W - \{v_1\}) \qquad (6)$$

$$\sum_{j \in V \cup W} z_{v_1 j} = 1 \qquad (7)$$

$$\sum_{j \in V \cup W} z_{jv_1} = \sum_{j \in V \cup W} y_j \qquad (8)$$

$$x_{ij} \leq z_{ij} \qquad (v_i, v_j \in V \cup W) \qquad (9)$$

$$x_{ij} \geq \frac{z_{ij}}{|W| + |V| + 1} \qquad (v_i, v_j \in V \cup W) \qquad (10)$$

$$y_k = 1 \qquad (\forall v_k \in C_t, \text{com } C_t \text{ definido em (2)}) \qquad (11)$$

$$y_k \in \{0,1\} \qquad (\mathrm{v_k} \in V \setminus T) \qquad (12)$$

$$x_{ij} \in \{0,1\} \qquad (v_i, v_j \in V \cup W) \qquad (13)$$

$$z_{ij} \in \mathrm{Z}^+ \qquad (v_i, v_j \in V \cup W) \qquad (14)$$

A restrição (4) assegura ao modelo que todos os vértices $v_k \in W$ estejam cobertos pela rota, enquanto a restrição (5) determina o grau dos vértices pertencentes `a solução. Sendo assim, todos os vértices da rota obrigatoriamente possuirão grau dois. As restrições (6), (7), (8) e (9), definem uma variável de fluxo viável z_{ij}, escoada na aresta $(i, j) \in E$. As restrições (10) e (11) asseguram a conectividade da rota e a restrição (12) garante a pertinência dos vértices $v_k \in T$ `a rota solução. Finalmente as restrições (13) e (14) definem as condições de integralidade do problema.

4.3. Testes de Redução para o Espaço de Busca

Uma importante parcela dos problemas técnico-científicos consiste em alguma forma de busca: dada uma certa coleção de elementos, deseja-se encontrar nela um ou mais elementos que atendam algumas condições previamente estabelecidas. O caso mais geral é a busca num certo conjunto, de uma solução, para um dado problema. De acordo com a sua generalidade, os algoritmos de busca podem ser classificados como fracos ou fortes. Quanto maior o conhecimento sobre o espaço de busca, mais especializados ficam os algoritmos para este espaço, aumentando a sua eficiência e perdendo a sua generalidade.

Em otimização, uma boa capacidade de exploração do espaço de busca é sempre requerida, uma vez que a finalidade é encontrar um extremo global ou pelo menos uma solução de boa qualidade. Diversos trabalhos nesta área tem utilizado uma técnica conhecida como redução do espaço de busca [22], [23], [24], cujo objetivo principal é a eliminação de informações que são irrelevantes ao estudo do problema e acabam por consumir mais tempo de execução nos algoritmos propostos para solucioná-lo.

Maniezzo [25] e Gendreau [15] apresentaram em seus trabalhos alguns testes para a simplificação do espaço de busca do PRR. Em virtude de estarmos considerando uma descrição mais genérica do PRR, a qual propomos a inclusão dos vértices $v_i \in W$ no conjunto dos vértices candidatos a composição da rota solução, os testes de redução existentes na literatura tiveram que ser adaptados ou substituídos por outros mais genéricos.

Seja $Cob = (cob_{ij})$ uma matriz $|W| \times |V \cup W|$, de zeros e uns (0-1), onde, $cob_{ij} = 1$,

$v_i \in W$, $v_j \in V \cup W$, se e, somente se, $c_{ij} \le d$. Podemos descrever os testes de redução do grafo do seguinte modo:

- Transformar em um vértice optativo $v_k \in V \setminus T$, qualquer vértice $v_i \in W$, tal que $cob_{ij} = 1, \forall j \in V$;
- Transformar em um vértice optativo $v_k \in V \setminus T$, todos os vértices $v_i \in W$, tal que $cob_{ik} = 1$, para $v_k \in T$;
- Remover de $V \setminus T$ todos os vértices v_i , tal que $cob_{ij} = 0, \forall v_j \in W$.

O primeiro e o segundo teste que estão sendo propostos, juntamente com o terceiro que já havia sido considerado em trabalhos anteriormente citados, formam o conjunto de testes de redução utilizados neste trabalho. Estes são aplicados uma única vez (na seqüência em que foram descritos) ao grafo da instância original do problema. Somente o grafo reduzido é utilizado posteriormente

Nas figuras a seguir, os testes são ilustrados passo a passo sob o grafo apresentado nas *Figuras 1* e *2*.

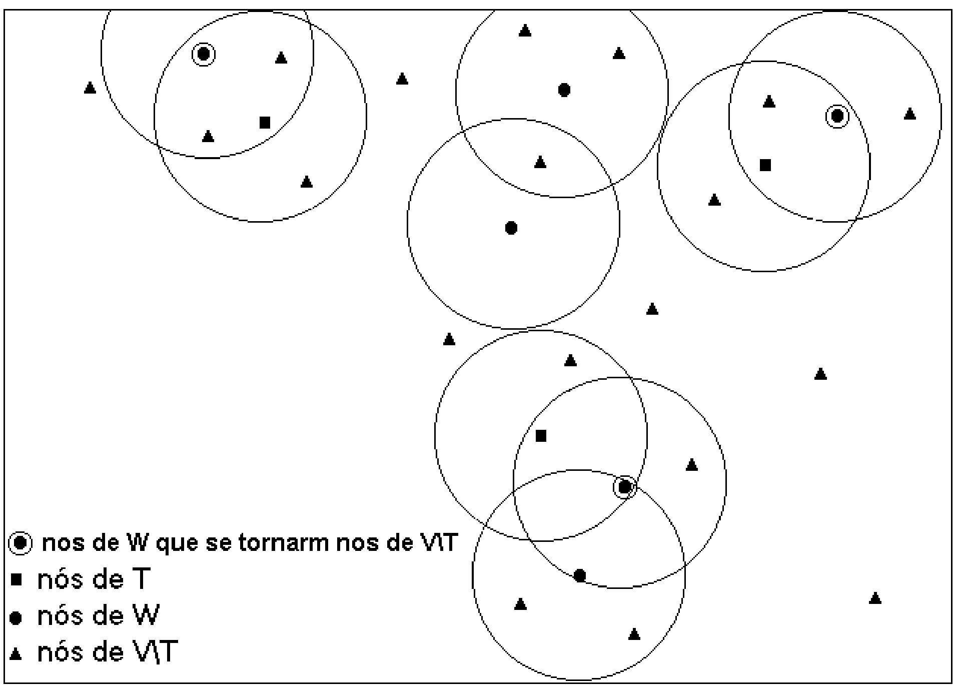

Figura 4: Ilustração do segundo teste de redução.

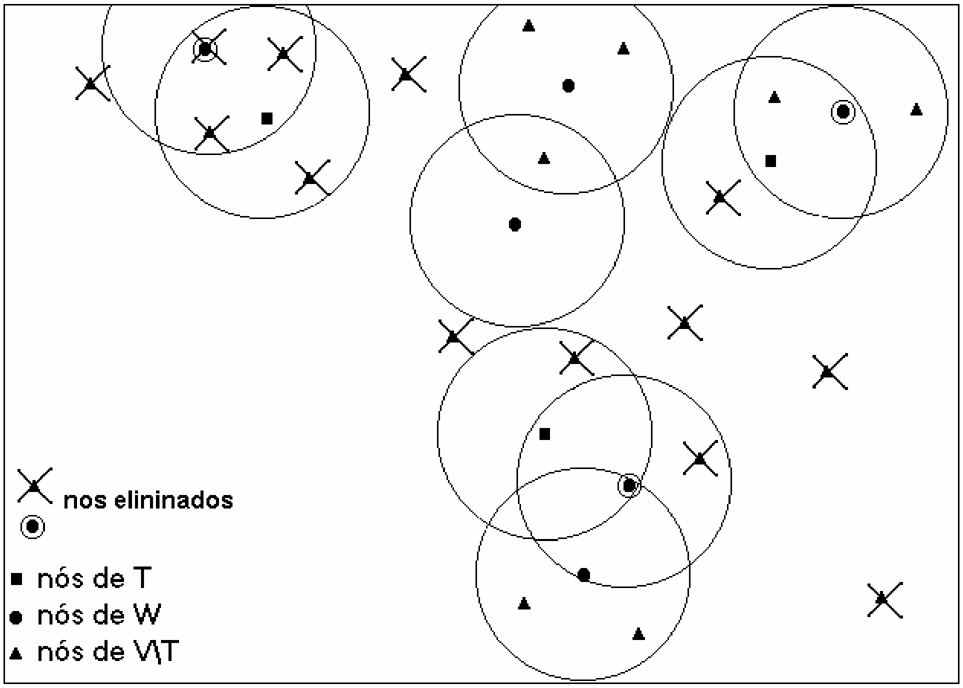

Figura 5: Ilustração do terceiro teste de redução.

4.4. Algoritmo de Construção - GRASP_SEQCLUST

Um fator importante a ser considerado, além da eficiência do algoritmo de busca, é a qualidade da solução inicial desta busca. Visando a obtenção de uma solução inicial viável, de boa qualidade e de suportável custo computacional, propomos uma metaheuristica, fundamentada nos conceitos da metaheuristica GRASP (*Greedy Randomized Adaptive Search Procedures*) desenvolvida no final da década de 1980 por Tom Feo e Maurício Resende [28]. Sua metodologia consiste em um processo iterativo onde cada iteração possui duas fases: uma fase de construção, na qual uma solução viável é construída, seguida de uma fase melhorias, cujo objetivo é encontrar uma solução ótima local.

A fase de construção é iterativa, adaptativa e pode ser gulosa. Ela é iterativa porque a solução inicial é construída elemento a elemento. É adaptativa porque os benefícios associados a cada elemento são levados de uma iteração para outra, refletindo as mudanças ocasionadas pela seleção prévia de outros elementos e pode ser gulosa porque a adição de cada elemento pode estar restrita a uma lista de apenas um candidato. Nesta fase, a escolha do próximo elemento a ser adicionado é determinado pela ordenação de todos os elementos candidatos em uma lista de candidatos C. Esta lista é construída de acordo com uma função gulosa $g : C \rightarrow \Re$, que mede o benefício de se selecionar cada elemento. O componente probabilístico do GRASP é caracterizado pela escolha randômica de um dos melhores elementos da lista de candidatos, não necessariamente o melhor.

A fase de melhorias consiste tipicamente em um procedimento de busca local, já que a solução gerada na fase de construção do GRASP pode não ser um ótimo global. De modo geral, é sempre benéfica a aplicação de uma busca local visando a melhoria da qualidade das soluções obtidas em uma fase de construção. Brevemente falando, um algoritmo de busca local consiste na substituição da solução corrente pela melhor solução da vizinhança desta. Na maioria dos casos a busca se encerra quando nenhuma solução melhor é encontrada na vizinhança.

O algoritmo GRASP_SEQCLUST para a obtenção de uma solução inicial viável para o PRR, também consiste de duas fases. A primeira fase realiza o sequenciamento do conjunto de todos os grupamentos do grafo (Nseq) e a segunda fase, encontra a melhor rota a partir dos grupamentos sequenciados na fase anterior.

A primeira fase se inicia com a escolha aleatória de um grupamento C_t definido por (2). Este será o primeiro grupamento da seqüência. Os demais serão escolhidos iterativamente a partir de uma lista de candidatos que estará diretamente relacionada ao último grupamento sequenciado. Se o tamanho desta lista for elevado, partimos para a construção de uma lista de candidatos restrita (LCR) a cada passo do algoritmo de construção. Quando não mais existirem grupamentos a serem sequenciados, o algoritmo iniciará a segunda fase.

A segunda fase tem a finalidade de definir que elemento (somente um) de cada grupamento pertencerá à rota. Adaptamos a idéia de Maniezzo et al. [25] para solucionar este problema.

Tendo a permutação de grupamentos $(C_t, C_k, ..., C_j, C_t')$ encontrada na primeira fase do algoritmo, onde C_t' é uma cópia do primeiro grupamento, cada caminho encontrado a partir desta permutação será uma solução viável para o problema. A melhor solução derivada de qualquer seqüência pode ser obtida computando o caminho mínimo a partir de um vértice do primeiro grupamento até a sua respectiva cópia associada ao grupamento duplicado. Utilizaremos o algoritmo de Dijkstra para encontrar este caminho.

Durante a fase de implementação, notamos que a complexidade para se obter a melhor solução a partir de uma permutação, poderia ser reduzida se utilizássemos como o primeiro grupamento da seqüência, um cuja cardinalidade fosse a menor possível. Partindo do fato de possuirmos uma lista circular, determinamos, sem perda de generalidade, sempre iniciar o sequenciamento da primeira fase com um grupamento definido pela equação (2), já que a cardinalidade deste tipo de grupamento é sempre 1.

No final da primeira fase, um grafo $G^* = (V \cup W, \overline{E})$ é gerado a partir do grafo original $G = (V \cup W, E)$. Em G^*, serão consideradas apenas as arestas entre os grupamentos adjacentes de acordo o sequenciamento obtido. Só então o algoritmo de Dijkstra é aplicado a G^*.

Como os grupamentos não são disjuntos, soluções inviáveis podem ser obtidas no final da segunda fase. Sendo assim, fez-se necessário a inclusão de um procedimento para a eliminação de subciclos. O critério adotado para este procedimento é eliminar os ciclos que apresentarem o maior acréscimo na rota.

4.5. Algoritmos de Busca Local - BUSCA_VNS

A busca local é a fase mais importante do processo e é também a mais dispendiosa computacionalmente. É nesta fase que vários ótimos locais são investigados com o objetivo de se atingir o ótimo global. Os principais aspectos responsáveis pelo sucesso de uma busca local consistem na escolha da estrutura de vizinhança, na eficiência da técnica de busca empregada nesta vizinhança e na qualidade da solução inicial.

Baseados na metaheuristica VNS (*Variable Neigborhood Search*) proposta por Hansen e Mladenovic em 1997 [16], [17], que consiste basicamente em uma sistemática troca de vizinhança associada a um algoritmo randômico de busca local, desenvolvemos nosso algoritmo de busca. Contrariamente a outras metaheuristicas baseadas em métodos de busca local, VNS não segue uma trajetória, mas explora incrementalmente vizinhanças distantes da solução corrente, atualizando esta solução somente quando uma melhora puder ser realizada. Neste caso, freqüentes características da solução atual são guardadas e utilizadas para a obtenção de soluções vizinhas promissoras. Uma outra característica importante do VNS é o uso de um conjunto finito N_k, $(k = 1,...,k_{\max})$, de estrutura de vizinhanças previamente selecionadas. A maioria das heurísticas de busca local utiliza apenas uma estrutura de vizinhança, podendo comprometer a qualidade da solução obtida caso a vizinhança seja inadequada.

Neste algoritmo de busca local utilizaremos uma seqüência de grupamentos associada à cada vértice da solução inicial, assim como a gerada ao final do algoritmo GRASPSEQCLUST. Partindo da idéia básica proveniente da metaheuristica VNS, realizaremos uma sistemática troca de vizinhanças, explorando incrementalmente as vizinhanças da solução corrente quando uma melhora puder ser realizada. A grande vantagem da utilização desta abordagem, é a utilização das características da solução atual para a obtenção de soluções vizinhas promissoras. Uma lista tabu esta sendo utilizada para evitar que a de busca retorne a soluções previamente visitadas.

As vizinhanças V_α de uma solução serão definidas do seguinte modo:

- $V_1 = \{\text{Efetuar uma permutação}\,(C_i, C_j),\, \text{com}\; i \neq j,\, \text{na solução atual}\};$
- $V_2 = \{\text{Efetuar duas permutação}\,(C_i, C_j)\;\text{e}\;(C_l, C_k),\, \text{com}\; i \neq k \;\text{e}\; j \neq l,\, \text{na solução atual}\};$
- $V_\alpha = \{\text{Efetuar}\;\alpha\;\text{permutação}\,(C_i, C_j), (C_l, C_k), \ldots (C_w, \mathrm{C}_z)\;\text{com os grupamentos diferentes ente s}i\};$

O valor de α é no máximo o número de grupamentos mais um, isto é, $(\alpha \leq (n^o\ de\ grupamentos) + 1)$.

Os grupamentos estão sendo selecionados aleatoriamente para a realização das trocas. Porém, um outro critério também está sendo estudado, onde consideramos uma lista de candidatos restrita associada a cada escolha de um grupamento.

A forma básica deste algoritmo pode ser apresentada da seguinte maneira:

Os algoritmos propostos foram implementados na linguagem C e os testes computacionais estão sendo realizados de duas maneiras. Para pequenas instâncias, o problema será resolvido de forma exata através de um pacote de métodos exatos conhecido como XPRESS. Os resultados serão comparados aos dos algoritmos propostos com o objetivo de se analisar a qualidade das soluções obtidas. Para grandes instâncias, exaustivos testes serão realizados para avaliar empiricamente os seus desempenhos.

4.6. Conclusões Parciais e Trabalhos Futuros

Apresentamos neste trabalho, uma nova formulação matemática para o Problema de Recobrimento de Rotas (PRR), intitulado originalmente por *The Covering Tour Problem* (CTP), descrito como um problema de programação linear inteira. Nesta formulação, propomos a utilização de variáveis de fluxo com o intuito de evitar a formação de rotas desconexas da origem. Do ponto de vista computacional, isso significa que trabalhos com um conjunto de restrições adicionais que crescem polinomialmente com o tamanho do problema, ao contrário das restrições tradicionais do Problema do Caixeiro Viajante para evitar subrotas, que crescem de forma exponencial com o tamanho do problema.

Esta formulação tem nos possibilitado encontrar a solução exata de pequenas instâncias do PRR utilizando o software XPRESS [29]. Com isso, será possível nas etapas seguintes do nosso trabalho, analisar a qualidade das soluções obtidas pelas metaheuristicas propostas confrontando-as com as soluções ótimas.

Um fator fundamental para viabilizar o uso de qualquer método, exato ou aproximado, na solução de problemas de elevada complexidade computacional, é o uso de técnicas para efetuar uma etapa de pré-processamento com o intuito de tentar reduzir o domínio do conjunto de soluções, eliminando informações irrelevantes na procura de um ótimo global para o problema. No nosso caso, como trabalhamos com uma generalização da descrição original do PRR (incluindo o conjunto W como candidato a composição da rota), os testes de redução existentes na literatura tiveram que ser adaptados ou substituídos por outros mais genéricos. Nesse sentido, outra contribuição de cunho teórico apresentada neste trabalho, se refere ao conjunto de testes redução para o modelo geral do

PRR adotado. Em simulações realizadas, observamos que o nível de redução do grafo associado ao problema, tem sido significativo reduzindo em alguns casos cerca de 40%.

Com o intuito de validar nossas propostas, estamos iniciando testes numéricos com um algoritmo adaptativo incluindo conceitos das metaheuristicas GRASP e VNS. A fase de construção de uma solução viável é obtida por um algoritmo fundamentado em alguns conceitos básicos da metaheuristica GRASP e a busca local é realizada pelo algoritmo do tipo VNS.

Referências Bibliográficas

[1] J. Current, Multiobjective Design of Transportation Networks, Ph.D. Thesis, Department of Geography and Environmental Engineering, The Johns Hopkins University, (1981).

[2] J. Current, E. Rolland, Efficient Algorithms for Solving the Shortest Covering Path Problem, Transportation Science 28 (1994), 317-327.

[3] J. Current, C. Revelle and J. Cohon, The Shortest Covering Path Problem: An Application of Locational Constraints to Network Design, Journal of Regional Sciences 24 (1984), 161-183.

[4] E.M. Arkin, M. M. Halldórsson and R. Hassin, Approximating the Tree and Tour Covers of a Graph, Information Processing Letters 47 (1993), 275-282.

[5] M. Hachicha, M. J. Hodgson and G. Laporte, Heuristics for the multi-vehicle covering tour problem, Computers and Operations Research 27 (2000), 29-42.

[6] J. Current and D. A. Schilling, The Median Tour and Maximal Covering Tour Problem: Formulations and Heuristics, European Journal of Operational Research 73(1994), 114-126.

[7] E. Balas, The prize collecting traveling salesman problem, Networks 19 (1989), 621-636.

[8] M. Fischetti and P. Toth, An additive approach for the optimal solution of the prize collecting traveling salesman problem, In Vehicle Routing: Methods and Studies, B.L. Golden and A.A. Assad (eds.), North-Holland, Amsterdam (1988), 319-343.

[9] G. Laporte and S. Martello, The Selective Traveling Salesman Problem, Discrete Appl. Math. 26 (1990), 193-207.

[10] E. M. Arkin and R. Hassin, Approximating Algorithms for the Geometric Covering Salesman Problem, Discrete Appl. Math. 55 (1994), 197-218.

[11] J. Current and D. A. Schilling, The Covering Salesman Problem}, Transportation Science 23 (1989), 208-213.

[12] J. R. Oppong and M. J. Hodgson, Spatial acessibility to health care facilities in suhum district, ghana, Professional Geographer 46 (1994), 199-209.

[13] M. Labbe and G. Laporte, Maximizing user convenience and postal service efficiency in post box location, Belgian J. Opns. res. Statist. and Computer Sci. 26 (1986), 21-35.

[14] C. S. Revelle and G. Laporte, New directions in plant locations, Studies in Locational Analysis 5 (1993), 31-58.

[15] M. Gendreau, G. Laporte, and F. Semet, The Covering Tour Problem, Operations Research 45 (1995), 568-576.

[16] P. Hansen and N. Mladenovic, Variable Neighborhood Search for the p-median, Location Science 5 (1997), 207-226.

[17] P. Hansen, N. Mladenovic and D. Perez-Brito, Variable Neighborhood Decomposition Search, GERAD (1998).

[18] M. Gendreau, A. Hertz and G. Laporte, New Insertion and Postoptimization Procedures for the Traveling Salesman Problem, Opns. Res. 40 (1992), 1086-1094.

[19] E. Balas and A. Ho, Set Covering Algorithms Using Cutting Planes, Heuristics, and Subgradiente Optimization: A Computacional Study, Math. Programming, 12 (1980), 37-70.

[20] CPLEX Optimizaton Inc., Using the CPLEX Callable Library and CPLEX Mixed Interger Library, (1993).

[21] M. Fischetti, J. J. Salazar and P. Toth, A Branch-and-Cut Algorithm for the Symetric Generalized Traveling Salesman Problem, Working Paper, DEIS, University of Bologna, (1994).

[22] C. W. Duin and A. Volgenant, Reduction Test for the Steiner Problem in Graphs, Networks 19 (1989), 549-567.

[23] P. Winter, Steiner Problem in Networks: A Survey, Networks 17 (1987), 129-167.

[24] A. E. Candia Vejar, O problema de Steiner em Grafos Dirigidos, PhD Thesis, Systems Enginering and Computer Sciences, COPPE, Federal University of Rio de Janeiro, Brazil (1991).

[25] V. Maniezzo, R. Baldacci, M. Boschetti and M. Zamboni, Scatter Search Methods for The Covering Tour Problem, Scienze dell'Informazione, University of Bologna, (1999).

[26] F. Glover, Scatter Search and Star Paths: Beyong the Genetic Metaphor, OR. Spektrum 17 (1995), 125-137.

[27] F. Glover, A Template for Scatter Search and Path Relinking, Lecture Notes in Computer Science, J.K. Hao, E. Lutton, E. Ronald, M. Schoenauer, D. Snyers (Eds.), (1997).

[28] M. G. C. Resende, and T. A. Feo, Greedy Randomized Adaptative Search Procedures, Journal of Global Optimization (1995), 1-27.

[29] XPRESS-MP Dash, Biswoth, Northants NN7 3BX, UK.