Estudo e Aplicação do Covering Tour Problem
Parte I

Aluno: Rodolfo Cunha Oliveira - RA:092891
Orientador: Prof. Dr. Antonio Carlos Moretti
Departamento de Matemática Aplicada – IMECC – UNICAMP

Sumário

1. Introdução e Motivação

Atualmente, um dos maiores problemas encontrados nas grandes cidades são aqueles relacionados ao transporte. O grande do número de veículos nas ruas, o aumento dos custos para manter e transitar com um veículo, bem como a necessidade de pontualidade dos serviços relacionados ao transporte são os principais fatores para a motivação do estudo e, portanto, do desenvolvimento de modelos matemáticos afim de resolver ou minimizar o problema.

Apesar da atualidade desta questão, o problema estudado é uma generalização de outro, proposto primeira vez no século XIX, pelos matemáticos W.R. Hamilton e T.P. Kirkman, o “Traveling Salesman Problem”(TSP) ou “Problema do Caixeiro Viajante”, que basicamente consiste em determinar o menor percurso dentre um conjunto de cidades, partindo de qualquer uma delas, visitando todas as cidades apenas uma vez e finalmente regressando à cidade inicial.

O TSP teve sua forma geral bem posta e profundamente estudada pela primeira vez em 1930, em Havard e Viena e ganhou força nas décadas de 1950 e 1960 quando foi expressado pela primeira vez como um problema de programação linear inteira pelos cientistas George Dantzig, D.R. Fulkerson e Selmer M. Johnson. [1]

Entretanto, entre as décadas de 1980 e 1990, os matemáticos J.R. Current e D.A. Schiling propuseram e estudaram uma nova generalização doTSP, um problema de maior porte, o Covering Tour Problem(CTP). [2]

O CTP consiste em traçar uma rota que visitará um conjunto de pontos obrigatórios e cobrir com uma margem acessível outros pontos obrigatórios.

Considerando os pontos como localidades e a rota, um trajeto a ser percorrido, é possível facilmente encontrar uma conexão entre esse problema e alguns dos problemas de transporte público e particulares atualmente.

Um bom exemplo para o CTP é o problema de determinar quais os melhores locais que um ônibus particular deve visitar afim de gerar a menor rota e cobrir todos os bairros relevantes para uma empresa, dado um conjunto obrigatório inicial de localidades que o ônibus deverá visitar e que este deverá sair e retornar para a empresa a cada ciclo.

Outro exemplo seria determinar a melhor rota que um segurança deve realizar afim de cobrir um conjunto de pontos importantes, como por exemplo quarteirões, conjuntos de prédios e casas, dado um conjunto obrigatório inicial de localidades que o segurança deverá visitar, tais como caixas eletrônicos e bancos, e que este deverá sair e retornar para uma base a cada ciclo.

A seguir, o problema será proposto e será apresentado um modelo matemático que busca resolvê-lo de maneira exata o problema.

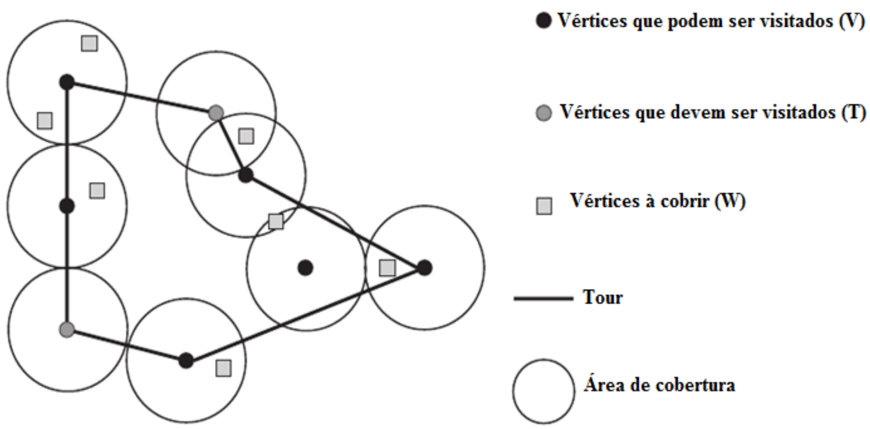
2. Teoria, Modelo

2.1 Teoria e modelo matemático do CTP:

Definição 1:

Seja $G = (V \cup W, E)$ um grafo desorientado e completo onde $V \cup W$ é o conjunto de vértices $V = \{v_1, \dots, v_n\}$ e $E = \{(v_i, v_j): v_i, v_j \in V \cup W, i < j\}$ é o conjunto de arestas. O vértice v_1 é um depósito, V é um conjunto de vértices que *podem* ser visitados, $T \subseteq V$ é o conjunto de vértices que *precisam* ser visitados, $(v_1 \in T)$ e W é o conjunto de vértices que precisam ser *cobertos.* A matriz distancia $C = (c_{ij})$ satisfaz a diferença triangular definida em E. O CTP consiste em determinar o menor comprimento de um tour ou ciclo Hamiltoniano sobre um conjunto V de modo que o tour contenha todos os vértices T e todos os vértices de W estejam cobertos pelo tour, ou seja, cada vértice de W está à uma distancia c de ao menos um vértice do tour. Tal tour, pode nem sempre existir. [2]

O problema pode ser ilustrado na figura 1:



[3]

Figura 1: Exemplo do Covering Tour Problem

Formução exata conhecida para o CTP

O CTP pode ser formulado como um problema de programação linear inteira:

Para cada $v_k \in V$, seja y_k uma variável binária igual a 1 se e somente se o vértice v_k pertence ao tour. Se $v_k \in T$, então y_k é necessariamente igual a 1. Para $v_i, v_j \in V$ e $i < j$, seja x_{ij} uma variável binária igual a 1 se e somente se a aresta (v_i, v_j) pertence ao tour. Também defina coeficientes binários δ_{lk} iguais a 1 se e somente se $v_l \in W$ pode ser coberto por $v_k \in V$, (ou seja $c_{lk} \leq c$), e seja $S_l = \{v_k \in V | \delta_{lk} = 1\}$ para vada $v_l \in W$. Assumamos que $|S_l| \geq 2$ para todo $v_l \in W$e que o tour degenarado (v_1) é infactível. [2]

O CTP, portanto pode ser formulado como:

$$Minimizar \sum_{i<j} c_{ij} x_{ij}, \tag{1}$$

Sujeito à:

$$\sum_{v_k \in S_l} y_k \geq 1 \; (v_l \in W), \tag{2}$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2y_k (v_k \in V), \tag{3}$$

$$\sum_{\substack{v_i \in S,\ v_j \in V \setminus S \\ ou\ v_j \in S,\ v_i \in V \setminus S}} x_{ij} \geq 2y_t \quad (S \subset V, 2 \leq |S| \leq n-2, T \setminus S \neq \emptyset, v_t \in S), \tag{4}$$

$$x_{ij} \in \{0,1\} \quad (1 \leq i < j \leq n), \tag{5}$$

$$y_k = 1 \;\; (v_k \in T), \tag{6}$$

$$y_k \in \{0,1\} \;\; (v_k \in V \setminus T). \tag{7}$$

Comentários sobre o modelo

A função objetivo:

$$Minimizar \sum_{i<j} c_{ij}x_{ij}, \qquad (1)$$

Notamos que o objetivo deste modelo é minimizar a soma dos c_{ij} ativos (aqueles que multiplicam os $x_{ij} \neq 0$), ou seja, minimizar o custo total do tour, este que normalmente é interpretado como a distância total.

As restrições:

$$\sum_{v_k \in S_l} y_k \geq 1 \ (v_l \in W), \qquad (2)$$

Este conjunto de restrições garante que todos os vértices $v_l \in W$, sejam cobertos por ao menos um vértice v_k. Ou seja, que cada um dos vértices que precisam ser cobertos, sejam efetivamente cobertos.

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2y_k(v_k \in V) \qquad (3)$$

O conjunto (3) força que, caso o vértice v_k esteja no Tour (ou seja, $y_k = 1$) então exatamente duas arestas ligadas à ele, também pertencerão ao tour.

$$\sum_{\substack{v_i \in S,\ v_j \in V \setminus S \\ ou\ v_j \in S,\ v_i \in V \setminus S}} x_{ij} \geq 2y_t \quad (S \subset V, 2 \leq |S| \leq n-2, T \setminus S \neq \emptyset, v_t \in S) \qquad (4)$$

Em (4), o grupo de restrições evita que a solução contenha sub-tours, ou seja que a resposta do problema seja um ciclo único e fechado. Para tal, as restrições devem ser criadas de tal maneira que para vértice existente no tour, exista um meio de chegar em qualquer outro vértice também pertencente ao tour.

$$x_{ij} \in \{0,1\} \quad (1 \leq i < j \leq n) \qquad (5)$$

Estas restrições determinam que as variáveis que representam as arestas (x_{ij}) sejam binárias.

$$y_k = 1 \ \ (v_k \in T) \qquad (6)$$

O grupo (6) determina que os vértices que pertencem ao tour inicial estejam no tour final.

$$y_k \in \{0,1\} \ \ (v_k \in V \setminus T) \qquad (7)$$

Por fim, este último conjunto tais restrições determinam que as variáveis que representam os vértices, sejam binárias.

2.2 O software GLPK

O software GNU Linear Programming Kit GLPK é um software-ferramenta desenvolvido para resolver problemas de larga escala de Programação Linear e Programação Linear Inteira Mista. Ele é uma ferramenta gratuíta e, por tal motivo, é comumente utilizado tanto por empresas de pequeno porte, quanto por estudantes que buscam iniciar a programação matemática.

O software GLPK utiliza o método do simplex revisado e o método de pontos interiores primal-dual para resolver problemas não inteiros e o algoritmo de branch-and-bound junto ao método dos planos de corte de Gomory para resolver os problemas inteiros (mistos) por meio do solver GLPSOL.

3. Metodologia e Implementação

As restrições do tipo (4) apresentam um grande problema de implementação, devido ao número exponencial de restrições existentes nesse grupo, fato que normalmente torna o CTP inviável de ser resolvido, portanto é necessário que exista uma abordagem heurística durante a implementação desse grupo.

Neste projeto, um algoritmo baseado no capítulo 3 da bibliografia [4] foi utilizado. Inicialmente, ele consiste em resolver o modelo proposto anteriormente através do software GLPK sem as restrições que evitam subtours.

Após obter uma solução, uma análise é realizada afim de determinar quais subtours ocorreram e determinar quantas e quais partições de vértices aqueles subtours formam.

A etapa seguinte consiste em adicionar restrições ao problema que evitem uma ou mais partições por completo, ou seja, não impedir apenas o subtour encontrado, mas todas as possíveis soluções que envolvam combinações daqueles vértices.

Finalmente, o problema é resolvido novamente pelo GLPK, e o algoritmo se repete até que a solução obtida contenha apenas uma partição, ou seja não contenha subtours.

O limite estipulado para a criação de restrições foi de no máximo 10 mil restrições por iteração

4. Resultados Obtidos

Com o objetivo de testar a a consistência do modelo matemático e da heurística propostos dois problemas foram criados. Cada problema contém um conjunto de vértices aleatoriamente dispostos num plano. Estes vértices estão distribuídos em três conjuntos, o conjunto de vértices que devem ser cobertos W, o conjunto de vértices que podem ser visitados V e o conjunto de vértices que devem ser visitados $T \subseteq V$.

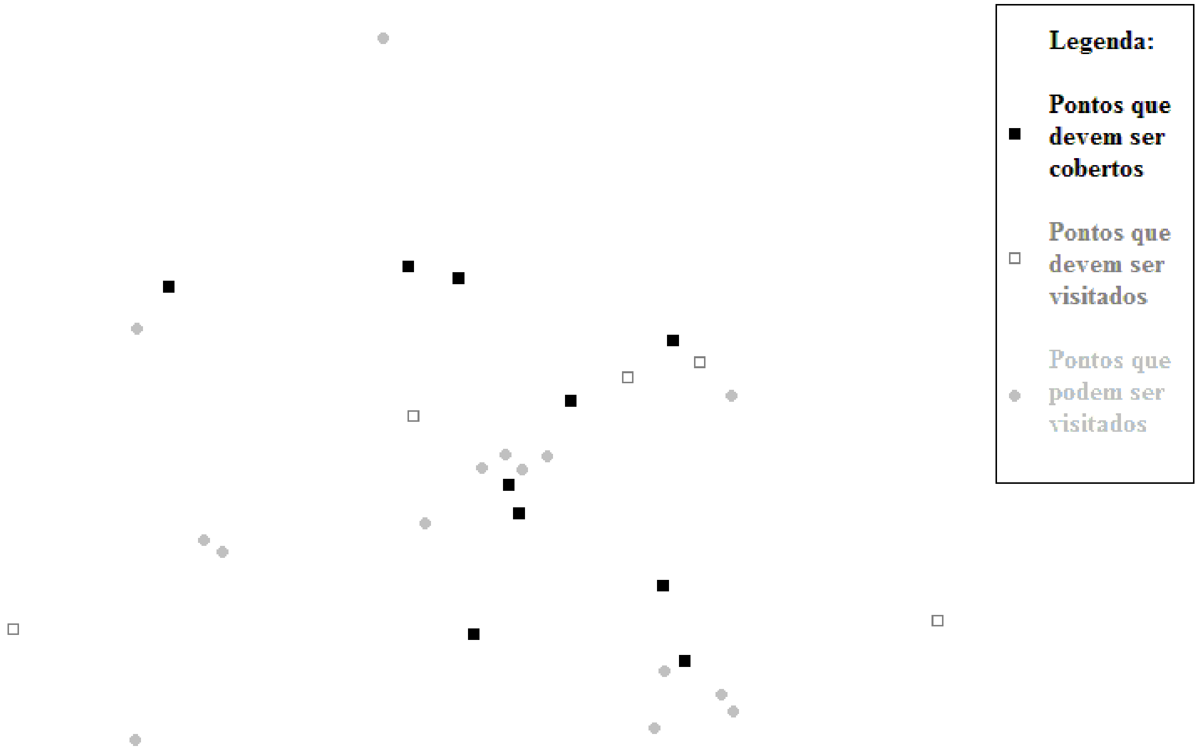


Figura 2: Esta figura retrata o Problema 1, este problema contém 10 vértices que devem ser cobertos, 5 vértices que devem ser visitados e 15 vértices que podem ser visitados, ou seja, o problema possui 30 vértices no total.

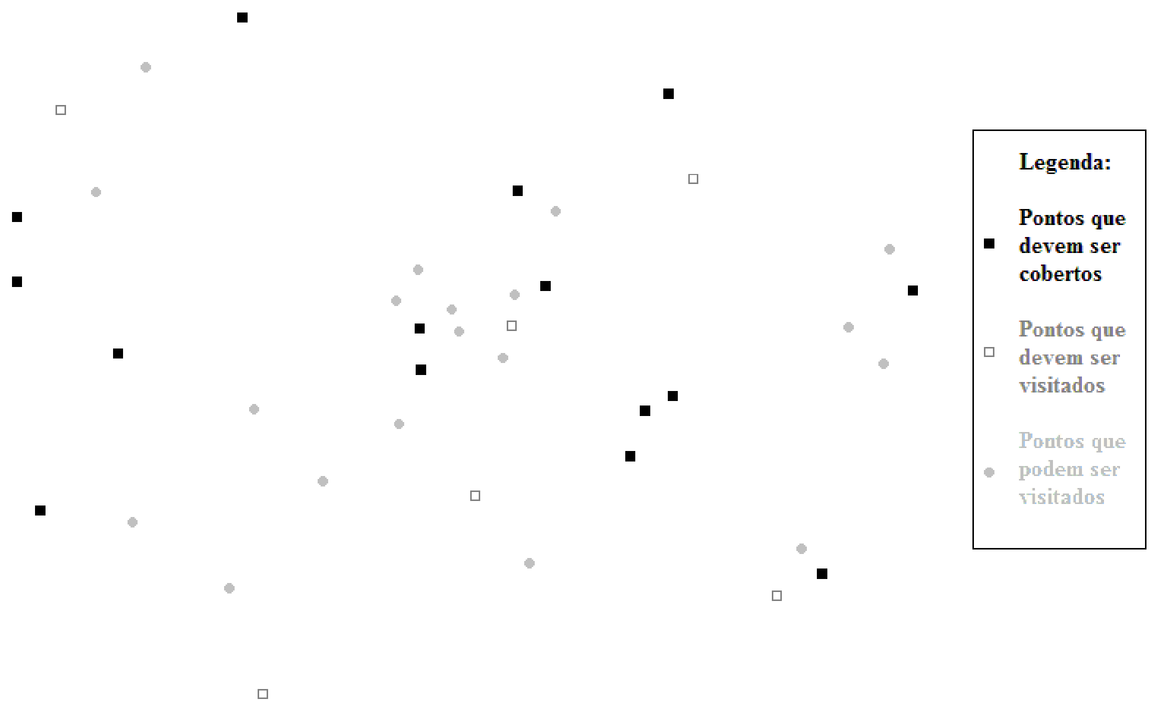


Figura 3: Esta figura retrata o Problema 2, este problema contém 15 vértices que devem ser cobertos, 6 vértices que devem ser visitados e 19 vértices que podem ser visitados, ou seja, o problema possui 40 vértices no total.

A Tabela 1 resume todas informações relevantes dos três problemas:

Tabela 1: Dados dos conjuntos de vértices dos três problemas

Problema	Vértices em W	Vértices em V/T	Vértices em T	Vértices em G
1	10	15	5	30
2	15	19	6	40

Através do procedimento proposto na metodologia, os dois problemas foram resolvidos com sucesso. Os resultados serão expressos através de tabelas que contém informações referentes ao tempo de execução do GLPK por iteração, o número total de restrições naquela iteração, o valor da função objetivo naquela iteração e o número de partições em cada iteração. Além das tabelas, a solução de cada problema também será ilustrada em uma figura, gerada pelo MATLAB.

Tabela 2: Resultados obtidos no Problema 1				
Iteração	**Tempo (em s)**	**# de restrições**	**# de partições**	**Valor da F.O.**
1	0.4	4986	3	392.5
2	0.4	5058	3	413.6
3	0.6	5130	2	415.4
4	0.7	5178	2	418.4
5	1.4	5658	1	423.9

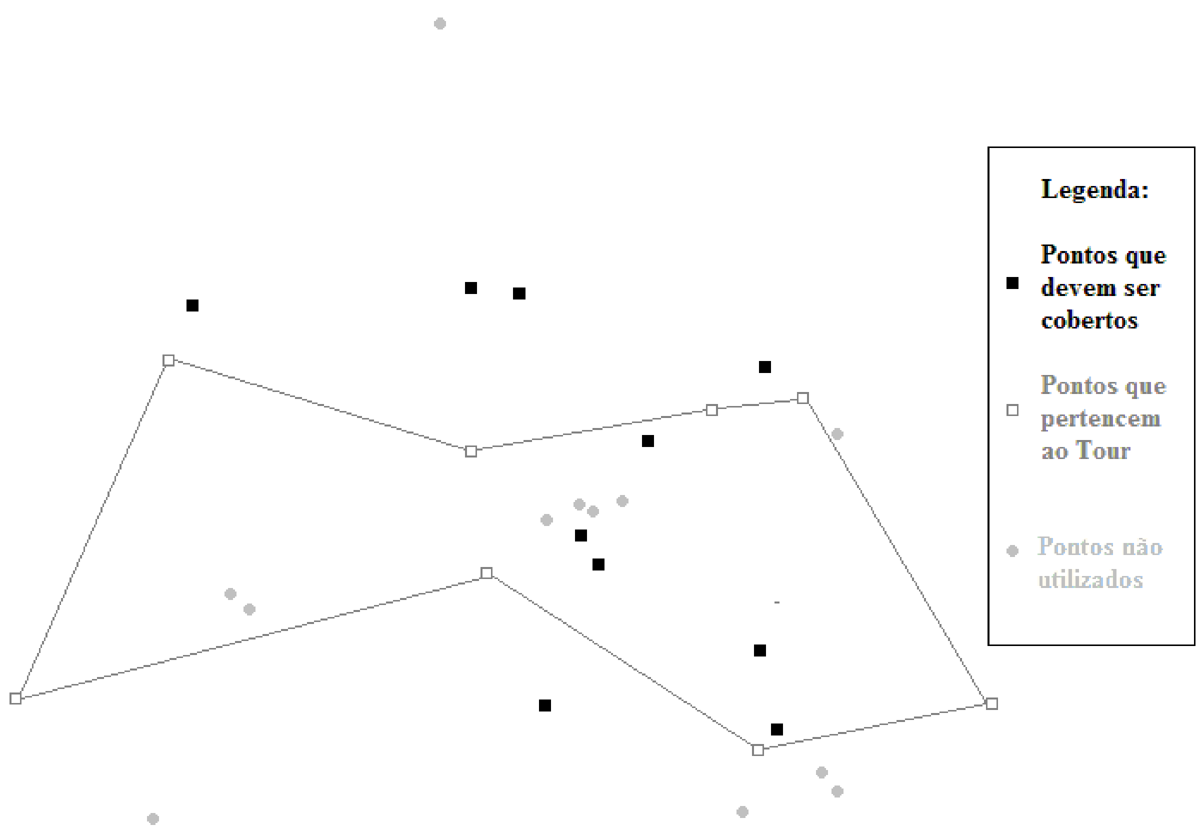


Figura 4: Ilustração da rota ótima que minimiza o problema 1 (7 vértices).

Tabela 3: Resultados obtidos no Problema 2				
Iteração	**Tempo (em s)**	**# de restrições**	**# de partições**	**Valor da F.O.**
1	2.1	11517	2	589.7
2	2.1	11541	2	590.6
3	2.2	11565	2	591.8

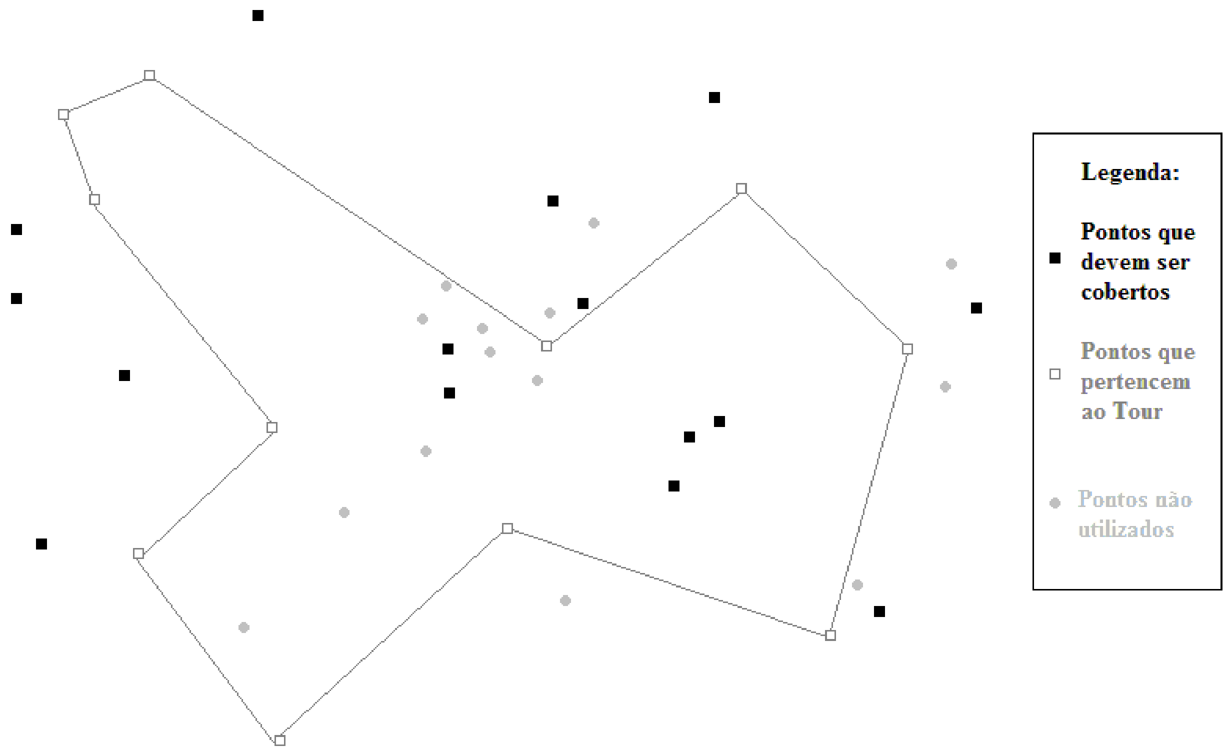


Figura 5: Ilustração da rota ótima que minimiza o problema 2(11 vértices).

5. Discussão e Considerações Finais

Os resultados demonstrados anteriormente mostram a eficácia do modelo matemático e do algoritmo utilizados para resolver o CTP. É importante notar o crescimento rápido de restrições adicionadas ao problema de um problema para outro. Entretanto, podemos notar que através da metodologia adotada, o número de restrições é relativamente pequeno quando comparado ao total de restrições que deveriam ser utilizadas na modelagem exata do problema.

Tabela 5: Comparação entre o número estimado de restrições e o número utilizado

Problema	Restrições Utilizadas	Restrições Estimadas
1	5658	Mais de 3.000.000
2	11565	Mais de 400.000.000

Assim, o objetivo final do projeto foi atingido e, motivado por este resultado, podemos sofisticar ainda mais o problema. Por exemplo, para um futuro trabalho utilizar alguma ferramenta mais sofisticada para a resolução de problemas de maior porte.

6. Bibliografia e Referências

[1] Wikipedia: http://en.wikipedia.org/wiki/Travelling_salesman_problem

[2] **The Covering Tour Problem** Michel Gendreau; Gilbert Laporte; Frédéric Semet *Operations Research*, Vol. 45, No. 4. (Jul. - Aug., 1997), pp. 568-576.

[3] **The bi-objective covering tour problem** Nicolas Jozefowieza, Frédéric Semetb, El-Ghazali Talbia.

[4] **Teaching Integer Programming Formulations Using theTraveling Salesman Problem** Gábor Pataki