

Problema do Caixeiro Viajante Probabilístico (ou com clientes estocásticos)

Renato L. F. Cunha

2008

Sumário

Sumário	i
1 Introdução	1
1.1 Motivação	1
1.2 Aplicações	2
2 Revisão bibliográfica	3
3 Problema do caixeiro viajante probabilístico	3
3.1 Tamanho esperado de um <i>tour</i>	3
3.2 Relação entre o TSP e o PTSP	4
4 Modelos	5
4.1 Definição do problema	5
4.2 Modelo para o TSP	6
4.3 Modelo de programação não linear para o PTSP	7
4.4 Modelo de programação linear para o PTSP	7
4.5 Modelo de programação inteira para o PTSP	8
4.6 Discussão dos métodos exatos	8
4.7 Branch and bound para resolução do PTSP	10
4.8 Um caso especial	11
5 Heurísticas	11
5.1 Heurística de <i>spacefilling</i>	12
5.2 Algoritmos de busca local	13
5.3 Aproximações da função objetivo	15
5.4 Otimização por Colônia de Formigas	17
5.5 Estado da arte – pACS+2.5-opt-EEais	18
6 Exercícios Propostos	20
Referências bibliográficas	21
A Implementação e experimentação	23
A.1 Metodologia	23
A.2 Experimentos	24
A.3 Conclusões e trabalhos futuros	24
B Soluções dos exercícios propostos	25

Agradecimentos

Agradeço a Rafael Ferreira Barra de Souza por suas excelentes contribuições com este trabalho. Em particular, seu método proposto para avaliação de soluções através de simulações garantiram maior correteza neste trabalho.

1 Introdução

Este documento trata sobre uma versão probabilística do problema do caixeiro viajante (ou o problema do caixeiro viajante com clientes estocásticos) e está estruturado da seguinte forma: neste capítulo é apresentada uma introdução sobre o problema, sua motivação e algumas aplicações. No capítulo 2 é apresentada uma breve revisão bibliográfica sobre o assunto, apresentando trabalhos seminais e principais artigos até a presente data. O capítulo 3 apresenta o Problema do Caixeiro Viajante Estocástico (*Probabilistic Traveling Salesman Problem*, ou PTSP) propriamente dito, com a definição formal do problema e alguns resultados básicos, como o relacionamento entre o PTSP e o TSP. No capítulo 4 são apresentadas técnicas exatas para solução do PTSP e uma discussão sobre os métodos exatos. Após a discussão do capítulo 4, o capítulo 5 apresenta abordagens heurísticas para o PTSP, apesar de o assunto não ser esgotado, o objetivo desse capítulo é apresentar os métodos mais usados na atualidade. O capítulo 6 apresenta uma breve lista de exercícios propostos para fixação de conceitos sobre o PTSP.

Os apêndices deste documento apresentam uma abordagem de implementação e experimentação para o problema do PTSP e respostas para alguns exercícios propostos.

1.1 Motivação

Assuma que uma companhia deseja encontrar um *tour* através de n clientes e deseja minimizar o custo de roteamento apenas uma vez. Caso haja a necessidade de visitar todos os clientes diariamente, então a opção natural para este caso é resolver este problema usando o Problema do Caixeiro Viajante (*Traveling Salesman Problem*, ou TSP) correspondente.

Assuma, no entanto, que este *tour* deve ser utilizado por um período prolongado de tempo e, nessa perspectiva, o conjunto de consumidores a serem visitados diariamente pode variar. Além disso, assumo que a companhia não é capaz de reotimizar ou simplesmente não deseja reotimizar o *tour* diariamente. Nesse caso, o veículo entregador seguirá o *tour* pré-determinado diariamente e, quando necessário, simplesmente ignorará os clientes ausentes. Esse problema não é mais um TSP, já que o *tour* encontrado não deve apensar ser bom, possuindo um baixo custo de roteamento, quando todos os clientes estiverem presentes, mas se manter bem comportado quando alguns clientes estiverem fora do conjunto original.

Baseado no problema supracitado, é possível formular o seguinte problema genérico: considere um problema de roteamento através de um conjunto de n pontos. Para cada instância do problema, apenas um subconjunto, consistindo de k pontos ($0 \leq k \leq n$), devem ser visitados. Desejamos encontrar um *tour* a priori através de todos os n pontos. Em uma dada instância, os k pontos presentes serão visitados na mesma ordem em que aparecem no *tour* a priori. O problema de encontrar um *tour* a priori de comprimento mínimo, no sentido da esperança probabilística, é definido como PTSP.

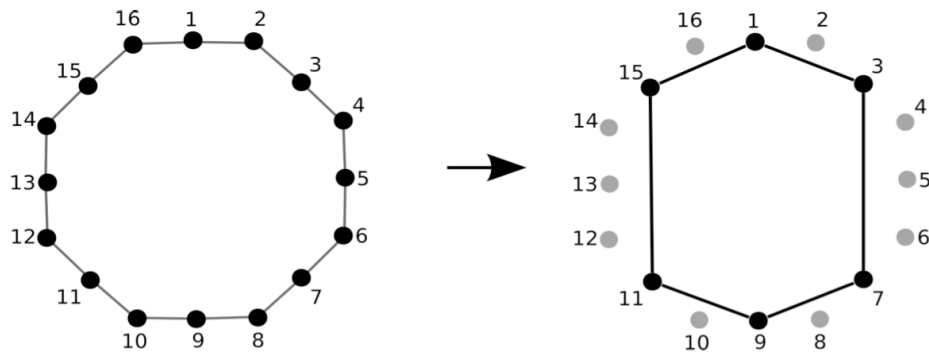


Figura 1: Um *tour* a priori para uma instância de 16 nós do PTSP onde os nós são visitados na ordem 1, 2, 3, ..., 15, 16 e 1. Assumindo que apenas os nós 1, 3, 7, 9, 11 e 15 devem ser visitados, a solução a posteriori visita os nós na ordem a priori, mas ignora os nós cinzas, que não devem ser visitados.

Como uma ilustração para o PTSP, pode-se considerar o caso de um carteiro que entrega suas cartas obedecendo a uma rota fixa. A cada dia, antes de realizar uma entrega em determinado lugar, ele verifica qual o próximo endereço deverá ser visitado e procede de acordo, ignorando endereços não presentes.

1.2 Aplicações

A seguinte discussão se encontra em [Bertsimas \[1988\]](#). O PTSP talvez seja o problema central na área de otimização combinatória probabilística. Sua importância advém de suas diversas aplicações e os problemas algorítmicos e combinatórios que apresenta.

Em geral, os PTSPs aparecem na prática quando uma companhia, em um dado dia, encontra o problema de coleções (entregas) de (para) um subconjunto de seu conjunto global de clientes em uma área e não deseja, ou simplesmente não pode, reprojeter seus *tours* diariamente.

Uma das razões para não reotimizar o *tour* para todas as instâncias do problema pode ser por que o operador do sistema não possui os recursos para fazê-lo. Ou pode ser decidido que essa nova otimização não é suficientemente importante para justificar o esforço e custo envolvidos. Ainda, um operador pode possuir outras prioridades, que podem ser melhor alcançadas mantendo a regularidade e personalização do serviço determinando que os mesmos veículo e motorista devam visitar um cliente em particular diariamente. Exemplos nessa categoria incluem serviços de entrega de alimentos [Bartholdi et al. \[1983\]](#), roteamento de empilhadeiras em um terminal de carga, ou a entrega diária de correio.

Em um contexto de planejamento estratégico, considere uma empresa de transportes que decidiu iniciar seu serviço em uma determinada área. A empresa realizou uma análise de mercado e identificou um número n de clientes em potencial que, durante qualquer período de coleta ou entrega, possuirão uma probabilidade considerável de necessitar de uma visita. Essa empresa deseja estimar os recursos necessários para servir a esses consumidores. Neste estágio do planejamento, a empresa atribui a mesma probabilidade p a todos os clientes potenciais que necessitam de uma visita. Para tratar do problema de planejamento, a empresa desejará estimar o tamanho de viagem esperado em um dia típico

para servir ao subconjunto dos n clientes que necessitam de uma visita.

2 Revisão bibliográfica

Este capítulo apresenta uma breve revisão bibliográfica sobre os trabalhos relacionados ao PTSP e, inevitavelmente, ao TSP.

O problema do PTSP foi introduzido por Jaillet [1985], que descreveu modelos matemáticos, limites e propriedades teóricas. Em particular, esse estudo demonstra que a solução do TSP determinístico pode ser arbitrariamente ruim para o PTSP. Outra propriedade interessante do PTSP é que em uma instância definida em um plano, a solução ótima pode cruzar consigo mesma, ao contrário do que acontece com o TSP [Flood, 1956]. Um conjunto de heurísticas usando o critério de vizinho mais próximo e um critério de *savings* foram implementados e testados por Jézéquel [1985] e por Rossi and Gavioli [1987]. Mais tarde, Bertsimas [1988] e Bertsimas and Howell [1993] investigaram algumas das propriedades dos PTSPs e propuseram novas heurísticas e limites para elas. Essas incluem *space filling curves* [Bartholdi and Platzman, 1982], uma implementação probabilística da *2-opt* e mudanças de vértices dentro de um *tour*. Posteriormente, Laporte et al. [1994] aplicou o método *Integer L-Shaped* ao PTSP, conseguindo resolvê-lo para até 50 vértices¹.

Em Bianchi and Knowles [2002], os limites obtidos em Bertsimas and Howell [1993] são provados errados e posteriormente corrigidos em Bianchi et al. [2005]. Devido à dificuldade de obter soluções exatas para o PTSP Jaillet [1985], a maioria dos esforços para solução deste problema se encontra em soluções heurísticas. Por exemplo, Bowler et al. [2002] usa uma variante de *Simulated Annealing*. Outros trabalhos usam Otimização por Colônia de Formigas (*Ant Colony Optimization*, ou ACO), uma heurística bem comportada para o TSP [Dorigo and Gambardella, 1997] e outros problemas de otimização combinatória, foi aplicada ao PTSP em Bianchi et al. [2002a], Bianchi et al. [2002b], Branke and Gunttsch [2003] e Branke and Gunttsch [2004].

Atualmente, o algoritmo que representa o estado da arte na solução do PTSP é baseado em ACO e é apresentado em Balaprakash et al. [2008]. Esse algoritmo é baseado em Bianchi et al. [2002a] e seu método de projeto é melhor descrito em Balaprakash et al. [2007a].

3 Problema do caixeiro viajante probabilístico

3.1 Tamanho esperado de um *tour*

Como supracitado, no PTSP, um *tour* sobre todos os pontos deve ser encontrado antes de saber quais pontos serão realmente visitados. Uma vez que um *tour* é encontrado, pontos ausentes serão simplesmente ignorados. Logo, para cada realização das variáveis aleatórias, cada uma correspondendo a um subconjunto dos pontos que realmente necessitam de visita, a ordem relativa dos pontos remanescentes no *tour* não será mudada. Um *tour*, por definição, possui um comprimento total aleatório. Portanto, a computação de

¹Em uma discussão em Jaillet [1985], estimou-se que, para modelos lineares deste problema, a solução exata seria possível apenas para cerca de meia dúzia de nós.

seu tamanho esperado corresponde à estratégia de visitar seus pontos de acordo com a seqüência prescrita pelo *tour* a priori, com os outros pontos sendo ignorados.

Considere, agora, um *tour* dado. Uma forma de computar seu tamanho esperado seria enumerar todos os casos e, para cada um deles, obter o tamanho do *tour* em consideração. No entanto, há 2^n casos, pois cada ponto está presente ou ausente, independentemente dos outros $n - 1$ pontos, e se assumirmos que computar o tamanho de um *tour* dado por uma seqüência de k pontos envolve $k - 1$ passos, este método necessitará de

$$\sum_{k=1}^n (k-1) \binom{n}{k} = (n-1)2^n$$

passos. Ou, de forma mais precisa, a esperança de uma permutação π sobre $[0, \dots, n-1]$, com $R_i, i = 1, \dots, 2^n$ representando a realização i das 2^n possíveis e $L_{R_i}(\pi)$ representando o tamanho do *tour* π para a realização R_i pode ser descrita por

$$E(\pi) = \sum_{i=1}^{2^n} p(R_i) L_{R_i}(\pi). \quad (1)$$

No entanto, ao invés de usar a equação 1, que é $\mathcal{O}(n2^n)$, uma versão que usa apenas $\mathcal{O}(n^2)$ passos foi apresentada em Jaillet [1985] e Bertsimas [1988]. Sua idéia básica é calcular uma soma ponderada sobre todas as distâncias entre as cidades. Um peso para a distância d_{ij} entre as cidades i e j é igual à probabilidade de que a aresta correspondente, e_{ij} será incluída no *tour* final. Uma aresta e_{ij} é incluída no *tour* final se e somente se apenas todas as cidades entre i e j no *tour* do PTSP forem deixadas de fora da realização final. Note que, para PTSPs simétricos, onde $d_{ij} = d_{ji}$, e_{ij} também for efetivamente incluída no *tour* se todas as cidades de j a i forem omitidas, já que o *tour* é um ciclo. A equação derivada por Jaillet é exibida na equação 2. Para simplificação, considere $q_i = 1 - p_i$.

$$E(\pi) = \sum_{i=1}^n \sum_{j=i+1}^n d(i, j) p_i p_j \prod_{k=i+1}^{j-1} q_k + \sum_{j=1}^n \sum_{i=1}^{j-1} d(j, i) p_i p_j \prod_{k=j+1}^n q_k \prod_{k=1}^{i-1} q_k \quad (2)$$

Note que, para distribuições de probabilidade homogêneas, com $p_i = p$ e $q = 1 - p$ para todas as cidades i , a equação 2 é reduzida à equação 3.

$$E(\pi) = p^2 \sum_{i=0}^{n-1} \sum_{j=1}^{n-1} d(i, (i+j \bmod n)) q^{j-1} \quad (3)$$

3.2 Relação entre o TSP e o PTSP

Dada a equação para obter o tamanho esperado de um *tour*, é possível encontrar algumas propriedades sobre o PTSP, algumas delas são abaixo listadas [Jaillet, 1985; Bertsimas, 1988]:

Uma das perguntas mais fundamentais que podem ser feitas é como o PTSP se relaciona com o TSP. Em outras palavras, como um *tour* do TSP através de n pontos probabilísticos se comportariam como uma solução para o PTSP? Jaillet prova que potencialmente muito ruim. Apenas sob algumas condições muito especiais as soluções do PTSP e do TSP são idênticas, como, por exemplo, quando os n pontos se encontram sobre os vértices de

um polígono convexo. Em geral, no entanto, não se pode garantir essa estrutura, a não ser em problemas muito pequenos. Por exemplo, os seguintes resultados são válidos para o caso de haver n pontos com probabilidades $p_i = p, \forall i = 1, 2, \dots, n$ e suas distâncias obedecerem a desigualdade triangular.

- O TSP ótimo possui garantia de resolver otimamente o PTSP para problemas com apenas cinco pontos ou menos e apenas três pontos ou menos quando a matriz de distâncias não for simétrica. Os erros que podem resultar de usar um *tour* ótimo do TSP podem ser muito grandes, como sugerido pelo seguinte: denotando τ_p o *tour* ótimo do PTSP e por L_{TSP} o tamanho do *tour* ótimo do TSP, então:
- Sob a desigualdade triangular, se o número de pontos deterministicamente presentes for m^2 , então, para $m \geq 1$ e para $m = 0$ e n primo, temos as equações 4 e 5. Jailliet conjectura que ambas são verdadeiras ainda que $m = 0$ e n não necessariamente primo. O lado esquerdo da equação 5 representa o erro resultante caso o *tour* ótimo do TSP fosse usado para resolver o PTSP ao invés da solução ótima τ_p . À medida que p diminui, a qualidade do *tour* do TSP pode se tornar um substituto arbitrariamente ruim para τ_p e, em Jailliet [1985], há exemplos concretos onde isso acontece.

$$E(L_{\tau_p}) \geq p L_{TSP} \quad (4)$$

$$\frac{E(L_{TSP}) - E(L_{\tau_p})}{E(L_{\tau_p})} \leq \frac{1 - p}{p} \quad (5)$$

Alguns resultados adicionais foram obtidos para o PTSP que, sob a ótica de resultados semelhantes para o TSP, se mostram contra-intuitivos. São eles:

- Para um PTSP no plano Euclidiano, o *tour* ótimo do PTSP pode se interceptar. Este resultado é interessante pois contrasta com um dos primeiros resultados dos *tours* ótimos do TSP que, sob uma métrica euclidiana, não se interceptam. Como essa característica é aproveitada por heurísticas para o TSP, ela deve ser levada em conta antes de aplicar uma heurística do TSP para o PTSP.
- Uma abordagem em programação dinâmica para o PTSP falha em resolver o problema. A abordagem em programação dinâmica para TSP é simples, a ponto de alguém acreditar que ela é capaz de resolver o PTSP. No entanto, como não é possível decompor, de maneira direta, o PTSP em estágios diferentes, não é possível aplicar a programação dinâmica de maneira satisfatória.

4 Modelos

4.1 Definição do problema

O problema do PTSP pode ser definido como um grafo $G = V, E, C, P$, onde:

²Ou seja, a probabilidade destes m pontos estarem presentes no *tour* é 1

- $V = \{1, 2, \dots, n\}$ é o conjunto de vértices do grafo;
- $E = \{(i, j) | i, j \in V, i \neq j\}$ é o conjunto de arestas que ligam os V nós;
- $D = \{d_{ij} | i, j \in E\}$ é o conjunto de custos das arestas, ou seja, a matriz de distâncias das arestas;
- $P = \{p_i | i \in V, 0 \leq p_i \leq 1\}$ é o conjunto das probabilidades de visita de nós, onde p_i é a probabilidade de o nó i ser visitado.

O objetivo do PTSP é encontrar o ciclo hamiltoniano (*tour*) que passe por todos os vértices com o menor comprimento médio esperado.

4.2 Modelo para o TSP

Antes de apresentar modelos para o PTSP, convém apresentar um modelo matemático para o TSP, já que os modelos do PTSP serão baseados nesse. O TSP, então, pode ser representado pela equação 6 como função objetivo e as equações 7 a 10 como restrições³.

$$\text{minimize } \sum_{i=1}^n \sum_{j=1}^n d(i, j) x_{ij} \quad (6)$$

$$\sum_{i=1}^n x_{ij} = 1, \forall j \in N \quad (7)$$

$$\sum_{j=1}^n x_{ij} = 1, \forall i \in N \quad (8)$$

$$\sum_{i, j \in S} x_{ij}, \forall S \subset N \quad (9)$$

$$x_{ij} \in \{0, 1\}, \forall i, j \in N \quad (10)$$

Nessa formulação, a variável binária x_{ij} assume valor igual a 1 se o arco $(i, j) \in E$ for escolhido para integrar a solução e 0 em caso contrário. S é um sub-grafo de G , onde $|S|$ representa o número de vértices desse sub-grafo. Assumimos, implicitamente, que x_{ii} não existe (ou, para fins práticos, terá valor infinito) e que teremos $n(n - 1)$ variáveis inteiras 0–1 e $\mathcal{O}(n^2)$ restrições. A restrição 7 garante que o caixeiro viajante sai da cidade i apenas uma vez e a restrição 8 garante que o caixeiro viajante chega à cidade j apenas uma vez. A restrição 9 garante que qualquer *sub-tour* será eliminado. A restrição 10, por sua vez, garante que a variável x_{ij} será binária.

³Esta formulação foi proposta por Dantzig Fulkerson e Johnson

4.3 Modelo de programação não linear para o PTSP

Dado o grafo G definido para o PTSP, seja t um dado *tour* de G e sejam x_{ij} as variáveis de decisão definidas em 10. É possível expressar o tamanho esperado do *tour* pela equação 11. Para essa formulação, assumamos que o grafo possui $m \in M$ e $n \in N$ nós, sendo os nós do conjunto M os nós com probabilidade $p_i = 1$ e os pontos do conjunto N os nós cuja probabilidade $p_i < 1$.

$$E(L_t) = \sum_{i=1}^n \sum_{j=1}^n \sum_{r=0}^n \alpha_r d(i, j) y_{ij}^{(r)} \quad (11)$$

onde:

- $\alpha_r = \sum_{k=r}^{n-2} \left(\frac{C(n-2-r, k-r)}{C(n, k)} \right) p(W = n - k) \forall r \in \{0 \dots n - 2\}$;
- $C(n, k) = \binom{n}{k}$.

$$y_{ij}^{(0)} \equiv x_{ij} \quad \forall i, j$$

$$y_{ij}^{(r)} \equiv \begin{cases} x_{ij}^{(r)} & \text{se } 1 \leq i, j \leq n \\ \sum_{k=0}^r (r+1-k) x_{ij}^{(k)} & \text{se } n+1 \leq i, j < n+m, \forall r \in \{1, \dots, n-1\} \\ \sum_{k=0}^r x_{ij}^{(k)} & \text{caso contrário} \end{cases}$$

$$y_{ij}^{(n)} \equiv \begin{cases} \sum_{k=0}^n x_{ij}^{(k)} & \text{se } n+1 \leq i, j \leq n+m \\ 0 & \text{caso contrário} \end{cases}$$

$$x_{ij}^{(r)} = \sum_{k_1, \dots, k_r=1}^n x_{ik_1} x_{k_1 k_2} \dots x_{k_r j}, \quad \forall r \in \{1, \dots, n-1\}$$

Esta modelagem da função objetivo, aliada às restrições 7 a 10, representa um modelo não linear para o PTSP. O termo α_r representa uma fórmula fechada para o tamanho esperado de um *tour* como uma soma ponderada de $L_t^{(r)}$, que, por sua vez, representa o tamanho do *tour* t quando passar por r nós que estarão, com certeza, presentes no *tour*. O termo $p(W = n - k)$ assume que a distribuição de probabilidades dos vértices é binomial e, portanto, é possível obter seu valor exato.

4.4 Modelo de programação linear para o PTSP

O conjunto de variáveis $x_{ij}^{(r)}$ definido em 11 pode ser representado pelo conjunto de inequações descrito na equação 12.

$$x_{ik}^{(r-1)} + x_{kj} - 1 \leq x_{ij}^{(r)} \leq 1 + x_{ik}^{(r-1)} - x_{kj}, \quad 1 \leq k \leq n \quad (12)$$

Esta representação permite formular o PTSP como um programa linear inteiro misto, com as variáveis x_{ij} e $x_{ij}^{(r)}$ como variáveis de decisão. As restrições, além das presentes na formulação anterior, são as dadas pela equação 12 e a função objetivo ainda é dada pela equação 11.

Comparada à formulação não linear, essa nova formulação é linear, mas adiciona $n^2(n-2)$ novas variáveis contínuas e, para cada uma delas, há $2n$ restrições, ou seja, um total de $2n^3(n-2)$ novas restrições. Para essa formulação, há, no total, $\mathcal{O}(n^3)$ variáveis contínuas adicionais e $\mathcal{O}(n^4)$ restrições adicionais se comparadas à formulação original.

É possível obter uma formulação com menos restrições e que dará uma formulação linear inteira do PTSP.

4.5 Modelo de programação inteira para o PTSP

As variáveis definidas em 12 podem ser expressas pelo conjunto de desigualdades exibido na equação 14.

$$\begin{aligned}
 x_{ij}^{(r)} &\leq 1 - z_{ij}^{(r-1)} \\
 x_{ij}^{(r)} &\leq 1 + z_{ij}^{(r-1)} \\
 x_{ij}^{(r)} &\geq \alpha_{ij}^{(r)} + \beta_{ij}^{(r)} - 1 \\
 \alpha_{ij}^{(r)} &\geq \frac{1}{n} [1 + (n-1)z_{ij}^{(r-1)}] \\
 \beta_{ij}^{(r)} &\geq \frac{1}{n} [1 - (n-1)z_{ij}^{(r-1)}] \\
 x_{ij}^{(r)}, \alpha_{ij}^{(r)}, \beta_{ij}^{(r)} &\in 0, 1 \\
 \text{onde } z_{ij}^{(r-1)} &= \frac{(\sum_{k=1}^n k(x_{ik}^{(r-1)} - x_{ij}))}{(n-1)}
 \end{aligned} \tag{13}$$

A equação 14, em conjunto com as restrições 7 a 10, com variáveis de decisão $0 - 1$ $x_{ij}^{(r)}, \alpha_{ij}^{(r)}, \beta_{ij}^{(r)}$ formam um modelo de programação inteira pura para o PTSP.

4.6 Discussão dos métodos exatos

Baseado nos métodos descritos nas seções anteriores, uma discussão sobre sua eficiência pode ser feita. Considere as seguintes definições:

F₁ Formulação não linear;

F₂ Formulação linear;

F₃ Formulação linear inteira pura.

A tabela 1 exibe a quantidade de variáveis e restrições para alguns valores de n para as diversas formulações. Como se pode ver por essa tabela, há pouca esperança de ser possível resolver problemas com mais de seis variáveis de forma exata para as formulações F₂ e F₃. Nessas formulações, o número de restrições e variáveis cresce muito rapidamente. Além do mais, para problemas desse tamanho, a enumeração completa pode ser uma alternativa melhor.

n	F ₁		F ₂		F ₃	
	# variáveis	# restrições	# variáveis	# restrições	# variáveis	# restrições
5	25	10	125	760	250	385
10	100	20	1000	18020	2500	4020
15	225	30	3375	87780	9000	18655
20	400	40	8000	288040	22000	36040
30	900	60	27000	1512060	76500	126060
50	2500	100	125000	12000100	362500	600100
100	10000	200				
200	40000	400				
250	62500	500				
300	90000	600				

Tabela 1: Tabela numérica exibindo o número de variáveis e restrições para as formulações não linear, linear e linear inteira.

No entanto, F_1 possui o mesmo conjunto de restrições que o TSP e essa é uma grande vantagem sobre F_2 e F_3 . Essas restrições foram analisadas por diversos autores e novos métodos de melhoria das restrições do TSP podem ser facilmente aplicados para a formulação F_1 . Além do mais, com as formulações F_2 e F_3 , perde-se uma vantagem importante, que é quando um conjunto viável de x_{ij} é obtido, as outras variáveis são todas determinadas automaticamente.

Em suma, um código de programação linear geral para as formulações F_2 e F_3 deveriam ser automaticamente descartados, pois os problemas resolvidos são tão pequenos que, ou um TSP pode resolvê-los de forma eficiente, ou a enumeração completa os resolveria⁴. Logo, é melhor que se concentre em algoritmos projetados especificamente para o PTSP.

4.7 Branch and bound para resolução do PTSP

Como dito anteriormente, a formulação F_1 é um problema de programação não-linear com uma função objetivo polinomial de ordem n nos x_{ij} e um conjunto de restrições lineares similares às do TSP.

Uma idéia geral pode ser expressada da seguinte forma, assumindo $X = (x_{ij})$: suponha que a função objetivo, $f(X)$, expressa em π possa ser decomposta em uma soma de uma função polinomial de baixa ordem, $g(X)$, e uma função $h(X)$, “menos importante” que $g(X)$. Suponha, também, que um limite inferior, h , esteja disponível e que \bar{X} seja uma solução viável para o problema. É possível, então, reduzir a região viável do problema limitando nossa busca por um *tour* ótimo do PTSP dentre os *tours* tais que $S = X : g(X) + h \leq f(\bar{X})$. Em outras palavras, todos os *tours* que obedecem à equação 14 devem ser descartados.

$$g(X) + h > f(\bar{X}) \quad (14)$$

Um procedimento de branch and bound poderia ser baseado na minimização de $g(X)$, uma tarefa certamente mais fácil que minimizar $f(X)$.

Fazendo $X = x_{ij}$, com x_{ij} as variáveis binárias previamente definidas e da equação π , é possível expressar a função objetivo como

$$L(X) = E(L_t) = \sum_{r=0}^{n-1} \alpha_r L^{(r)}(X)$$

onde $L^{(r)}(X)$ é um polinômio de ordem $r + 1$. Em [Jaillet \[1985\]](#), é provado que os pesos α_r formam uma seqüência não decrescente. Logo, dependendo da escolha da distribuição de probabilidades dos nós, apenas os primeiros termos de $L(X)$ serão os mais importantes.

Limitando a investigação para casos particulares onde $\alpha_0 L^{(0)}(X)$ é o termo dominante, acredita-se que são os casos que podem ser resolvidos por técnicas desenvolvidas para o PTSP para problemas de tamanho modesto⁵.

⁴Note que esta discussão se baseia quase exclusivamente nos resultados de [Jaillet \[1985\]](#) e, portanto, podem estar desatualizados. No entanto, devido aos poucos artigos encontrados que tratam sobre modelagens exatas do PTSP, é difícil dizer o quanto essa área avançou nos últimos anos, se é que avançou.

⁵É importante notar que estes comentários são comentários feitos em uma tese de 1985 e é provável que avanços nesta área dêem resultados - e algoritmos - melhores. No entanto, devido ao maior foco em

Quando os casos considerados necessitarem de polinômios de mais alta ordem, não será possível utilizar métodos exatos desenvolvidos para o TSP e novos métodos deverão ser considerados. Quando se considera que o TSP é um caso especial do PTSP, onde a distribuição de probabilidades é 1 para todos os vértices, e que foram necessárias décadas de estudos para desenvolvimento de métodos de solução do TSP, percebe-se a dificuldade dos estudos de métodos exatos no PTSP.

Algoritmo branch and bound para o PTSP

Nesta subseção é apresentado um esquema geral para solução do método branch and bound proposto anteriormente. Ele é apresentado em pseudocódigo no algoritmo 1, com as seguintes suposições:

α_0 peso de $L_{1,t}^0$;

E_i tamanho esperado do filho i ;

h limite inferior de $\sum_{r=1}^{n-1} \alpha_r L_{1,t}^{(r)}$.

Note que a estrutura básica deste algoritmo é baseada no algoritmo de branch and bound apresentado em [Papadimitriou and Steiglitz \[1998\]](#), que deve ser consultado para maiores esclarecimentos, se necessário. Esta seção encerra a discussão sobre métodos exatos para o PTSP.

4.8 Um caso especial

Baseados na equação de forma fechada para determinação do tamanho de um *tour* do PTSP (equação 2), é possível encontrar um caso especial do PTSP que pode ser resolvido em tempo polinomial, chamado PTSP constante.

Considere as probabilidades que regem a presença dos nós, p_i para todo nó i e assumamos que a matriz de distâncias seja da forma $d(i, j) = u_i + v_j$. Neste caso, é possível provar que todos os *tours* possuem o mesmo tamanho esperado. A nova equação da esperança é exibida na equação 15 e sua prova será omitida. Leitores interessados podem se referir à página 90 de [Bertsimas \[1988\]](#) para a prova desta proposição.

Se um nó i está presente com probabilidade p_i , qualquer *tour* τ possui o mesmo tamanho esperado:

$$E(L) = \sum_{i=1}^n (u_i + v_i) p_i \left(1 - \prod_{k \neq i} (1 - p_k)\right) \quad (15)$$

5 Heurísticas

Como é possível perceber, o PTSP é um problema difícil e, portanto, para problemas de tamanho médio a grande, parece razoável se concentrar em procedimentos que, apesar

heurísticas por parte dos pesquisadores e da indisponibilidade de artigos gratuitos onde métodos exatos são explorados, o autor não é capaz de dizer o quão próximos - ou distantes - estes comentários estão do estado da arte em termos de soluções exatas para o PTSP.

Algoritmo 1 Branch-and-bound para solução do PTSP

```
activeset  $\leftarrow$  {0}
U  $\leftarrow$   $\infty$ 
currentbest  $\leftarrow$  1
while activeset is not empty do
  choose a branching node, node  $k \in$  activeset
  remove node  $k$  from activeset
  generate the children of node  $k$ , child  $i$ ,  $i = \{1, \dots, n_k\}$ ,  $z_i$ 
  for all  $i = \{1, \dots, n_k\}$  do
    if  $z_i \geq \frac{U-h}{\alpha_r}$  then
      kill child  $i$ 
    else
      if child  $i$  is a tour then
        compute  $E_i$ 
        if  $E_i \leq U$  then
           $U \leftarrow E_i$ 
          currentbest  $\leftarrow$  child  $i$ 
        else
          add child  $i$  to activeset
        end if
      end if
    end if
  end for
end while
```

de não garantirem uma solução ótima, servirão como “boas” aproximações. Ao longo dos anos, algumas heurísticas baseadas em heurísticas para o TSP foram desenvolvidas e algumas metaheurísticas aplicadas ao PTSP. Algumas heurísticas serão apresentadas neste documento. Note que não é seu objetivo esgotar o assunto. Antes de tratar especificamente sobre heurísticas, breves comentários sobre limites serão feitos.

5.1 Heurística de *spacefilling*

A heurística *spacefilling* foi proposta em [Bartholdi and Platzman \[1982\]](#) para o TSP euclidiano e, originalmente, é um método $\mathcal{O}(n \log n)$ baseado em ordenação que funciona da seguinte forma:

1. Dadas as n coordenadas (x_i, y_i) dos pontos no plano, compute o número $f(x_i, y_i)$ para cada ponto. A função $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ é baseada na curva de Sierpinski⁶.
2. Ordene os números $f(x_i, y_i)$ e visite os pontos na mesma ordem, produzindo um *tour* τ_{SF} .

Considerando uma instância S do problema, supondo que a heurística produz um *tour* $\tau_{SF}(S)$ baseado nessa instância e considerando o *tour* τ_{SF} produzido pela heurística

⁶Para detalhes, consulte [Bartholdi and Platzman \[1982\]](#).

na instância original, [Bertsimas \[1988\]](#) prova que a heurística produz um *tour* τ_{SF} com a propriedade:

$$\frac{E(L_{\tau_{SF}})}{E(L_{\tau_P})} \leq \frac{E(L_{\tau_{SF}})}{E(\sum_{TSP})} = \mathcal{O}(\log n)$$

Onde \sum_{TSP} corresponde a encontrar o menor *tour* do conjunto de nós presentes em todas as instâncias.

5.2 Algoritmos de busca local

A busca local é uma das heurísticas mais eficientes para muitos problemas de otimização combinatória que é baseada no conceito de tentativa e erro. Os algoritmos de busca local executam repetidamente modificações de uma solução, aceitando uma modificação quando seu custo é negativo. De maneira mais formal, dada uma instância (F, c) de um problema de otimização, onde F é o conjunto de soluções viáveis e c é uma função de custo, escolhamos uma vizinhança

$$N : F \rightarrow 2^F$$

que sofre uma busca por melhorias no ponto $t \in F$ pela função

$$\text{melhore}(t) = \begin{cases} \text{qualquer } s \in N(t) \text{ com } c(s) < c(t), & \text{se tal } s \text{ existir} \\ \text{"não"}, & \text{caso contrário.} \end{cases}$$

Um algoritmo de busca local genérico é exibido no algoritmo 2 e seu funcionamento básico inicia em uma solução viável $t \in F$ e usa a função $\text{melhore}(t)$ para buscar por uma melhor solução em sua vizinhança. Se houver uma solução melhor, então ela é escolhida e uma nova busca por sua vizinhança é realizada, até que um ótimo local seja alcançado. Note que algumas decisões devem ser feitas, como qual o método para encontrar uma solução inicial, para que a busca local seja aplicada. No entanto, essa discussão está fora do escopo deste documento. [Papadimitriou and Steiglitz \[1998\]](#) e [Bianchi \[2003\]](#) fazem uma discussão mais profunda sobre o assunto.

Algoritmo 2 Algoritmo genérico de busca local

```

t ← alguma solução inicial em F
while (melhore(t) ≠ "não") do
  t ← melhore(t)
end while

```

As subseções seguintes apresentam os algoritmos 2-p-opt e 1-shift, apresentados em [Bertsimas \[1988\]](#); [Bertsimas and Howell \[1993\]](#).

2-p-opt

Dado um *tour* a priori λ , sua vizinhança 2-p-opt é dada pelo conjunto de *tours* obtidos pela reversão de uma seção de λ e o ajuste dos arcos adjacentes à seção revertida, como exibido na figura 2. Seu funcionamento básico é apresentado no algoritmo 3.

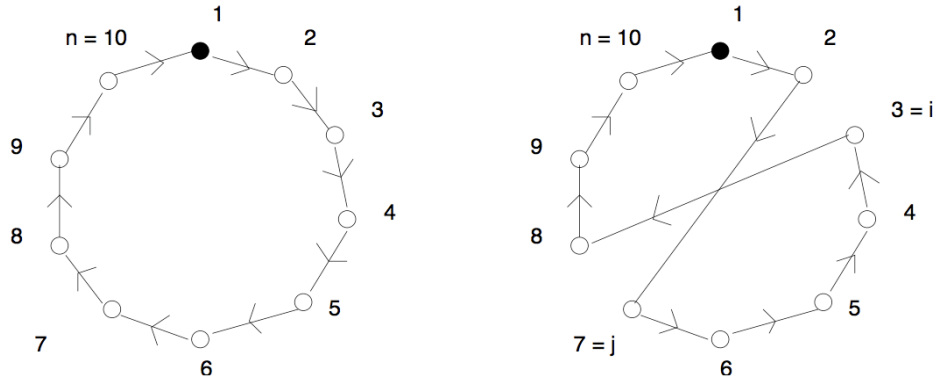


Figura 2: Tours $\lambda = 1, 2, \dots, i, i + 1, \dots, j, \dots, n$ (à esquerda) e $\tilde{\lambda} = 1, 2, \dots, i - 1, j, j - 1, \dots, i, j + 1, \dots, n$ (à direita) obtido de λ após reverter a seção $(i, i + 1, \dots, j)$, com $n = 10, i = 3, j = 7$.

Algoritmo 3 2-p-opt

Entrada: τ, i, j

$c \leftarrow 0$

$d \leftarrow 0$

for $k \leftarrow i$ to j **do**

$\text{tmp}_c \leftarrow \text{tmp}_k$

$c \leftarrow c + 1$

end for

for $k \leftarrow c - 1; k \geq 0; k \leftarrow k - 1$ **do**

$\tau_{i+d} \leftarrow \text{tmp}_k$

$d \leftarrow d + 1$

end for

1-shift

Dado um *tour* λ , sua vizinhança 1-shift é o conjunto de *tours* obtidos pela movimentação de um nó i para a posição j do *tour*, com os nós intermediários movidos uma posição para trás no *tour*, como exemplo da figura 3 e seu funcionamento básico é exibido no algoritmo 4.

Algoritmo 4 1-shift

Entrada: τ, i, j

$\text{tmp} \leftarrow \tau_i$

for $k \leftarrow i$ to j **do**

$\tau_k \leftarrow \tau_{k+1}$

end for

$\tau_j \leftarrow \text{tmp}$

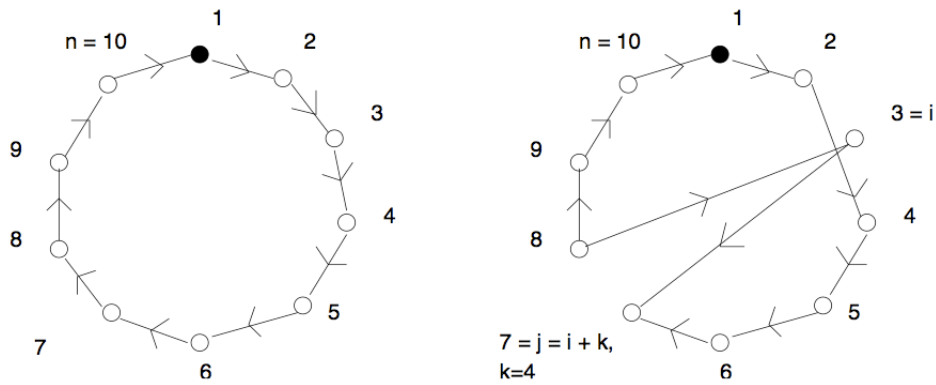


Figura 3: Tours $\lambda = 1, 2, \dots, i, i + 1, \dots, j, \dots, n$ (à esquerda) e $\tilde{\lambda} = 1, 2, \dots, i - 1, i + 1, i + 2, \dots, j, i, j + 1, \dots, n$ (à direita) obtido de λ após mover o nó i para a posição j e mover um espaço para trás os nós $i + 1, \dots, j$, com $n = 10, i = 3, j = 7$.

5.3 Aproximações da função objetivo

Apesar de o tempo de avaliação da esperança dado pela equação 2 ser polinomial, a complexidade de tempo resultante, $\mathcal{O}(n^2)$, para cálculo do tamanho esperado do *tour* ainda é muito grande, se comparada à complexidade de avaliação $\mathcal{O}(n)$, do TSP determinístico. Por esta razão, é conveniente explorar métodos de estimar de forma rápida o valor da função objetivo para tornar o processo de otimização mais rápido. Nesta seção são apresentados dois métodos de aproximação, o primeiro chamado aproximação por profundidade e, o segundo, de aproximação por amostragem.

Aproximação por profundidade

Diz-se que uma aresta possui profundidade $\theta \in [0, n - 2]$ com respeito a um *tour* a priori λ se existem exatamente θ cidades no *tour* entre i e j . Uma grande profundidade para uma aresta implica em uma baixa probabilidade da aresta fazer parte efetivamente do *tour* realizado, já que isso significaria que uma quantidade θ grande de consumidores não necessitariam de visita. Logo, arestas com profundidade maior deveriam impactar menos no valor do objetivo de um *tour*.

De maneira mais formal, dado um *tour* λ a priori e valores de profundidade $\theta = 0, 1, 2, \dots, n - 2$, as arestas consideradas numa instância do PTSP são agrupadas em conjuntos $\lambda^{(\theta)}$, cada um contendo arestas com profundidade igual. Note que o conjunto de arestas $\lambda^{(\theta)}$ possui um número de *sub-tours* iguais a $\text{mdc}^7(1, \theta + 1)$. Por exemplo, o conjunto $\lambda^{(0)}$ contém exatamente as arestas do *tour* a priori. A figura 4 exhibe alguns exemplos de agrupamentos de arestas baseados nas profundidades.

Dada a partição de arestas de acordo com sua profundidade $\theta = 0, 1, 2, \dots, n - 2$, a função objetivo do PTSP pode ser escrita como uma soma de termos de profundidade crescente, como apresentado na equação 16, onde $\lambda = (\lambda(1), \lambda(2), \dots, \lambda(n))$ é o *tour* a priori. Para o caso homogêneo, a equação 16 pode ser reduzida à equação 17.

⁷maior divisor comum

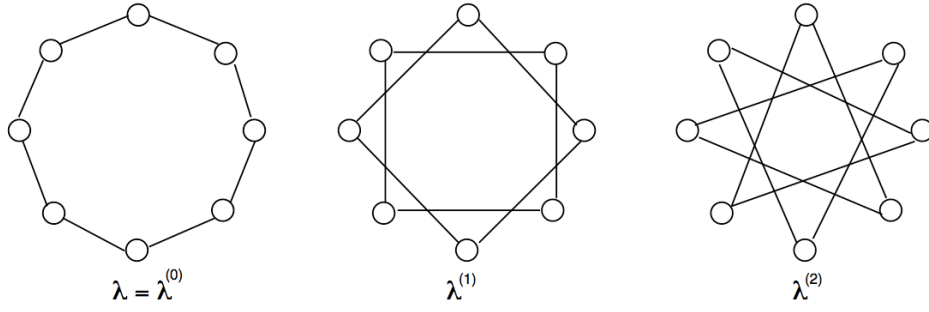


Figura 4: Arestas agrupadas de acordo com sua profundidade θ . A figura exhibe os conjuntos com $\lambda^{(\theta)}$, com $\theta = 0, 1, 2$. Note que $\lambda^{(0)}$ corresponde ao conjunto de arestas que formam o *tour* a priori.

$$E[L(\lambda)] = \sum_{\theta=0}^{n-2} \left[\sum_{j=1}^n d_{\lambda(j), \lambda(j+\theta+1)} p_{\lambda(i)} p_{\lambda(i+\theta+1)} \prod_{k=1}^{\theta} (1 - p_{\lambda(i+k)}) \right] \quad (16)$$

$$E[L(\lambda)] = \sum_{\theta=0}^{n-2} L(\lambda^{(\theta)}) p^2 (1 - p)^{\theta} \quad (17)$$

O que acontece, neste caso, é que a probabilidade de uma aresta de profundidade θ cai exponencialmente com o aumento de θ , logo é possível parar o processo iterativo após alguma profundidade fixa for alcançada, resultando na função objetivo

$$E^{\theta}[L(\lambda)] = \sum_{r=0}^{\theta} \left[\sum_{j=1}^n d_{\lambda(j), \lambda(j+r+1)} p_{\lambda(i)} p_{\lambda(i+r+1)} \prod_{k=1}^r (1 - p_{\lambda(i+k)}) \right] \quad (18)$$

que, simplificada para o caso homogêneo, se torna

$$E^{\theta}[L(\lambda)] = \sum_{r=0}^{\theta} L(\lambda^{(r)}) p^2 (1 - p)^r \quad (19)$$

A complexidade dessa função de aproximação é $\mathcal{O}(n\theta)$. Logo, quanto menor o valor de θ , maior o ganho de desempenho comparado à função exata. No entanto, quanto menor o θ , maior o erro entre a computação aproximada e a exata. Para mais detalhes sobre o método de aproximação por profundidade, incluindo avaliação da qualidade das aproximações, o leitor pode recorrer a [Bianchi \[2003\]](#).

Aproximação por amostragem

Na aproximação por amostragem, a computação da função objetivo é baseada no fato de que a função objetivo computa o valor esperado de uma quantidade aleatória. Logo, pode ser estimada por uma técnica de amostragem. Em outras palavras, a função objetivo pode ser escrita como

$$E[L(\lambda)] = \sum_{S \subset V} p(S) L(\lambda_S) \quad (20)$$

onde S é um subconjunto dos clientes V , $L(\lambda_S)$ é o comprimento do *tour* λ , podado de forma que apenas clientes em S sejam visitados e $p(S)$ é a probabilidade do subconjunto de clientes S necessitarem de visita:

$$p(S) = \prod_{i \in S} p_i \prod_{i \in (V-S)} (1 - p_i) \quad (21)$$

A função objetivo da equação 20 computa o valor esperado do *tour* λ sobre todos os subconjuntos aleatórios de clientes. A idéia, na aproximação por amostragem, é estimar o valor esperado da seguinte forma. O comprimento $L(\lambda)$ é uma variável discreta, com o valor $L(\lambda_S)$ e probabilidade $p(S)$. Sejam S_i , $i \in 1, 3, \dots, N$ subconjuntos dos n clientes originais amostrados com probabilidade $p(S_i)$. Se cada subconjunto S_i é amostrado independentemente dos outros, o teorema do limite central implica que

$$\frac{1}{N} \sum_{i=1}^N L(\lambda_{S_i}) \approx E[L(\lambda)], \text{ para } N \text{ suficientemente grande.} \quad (22)$$

A complexidade da aproximação por amostragem é $\mathcal{O}(nN)$ e, como no caso da amostragem por profundidade, quanto menor o N , mais rápida a avaliação e maior o erro.

Para mais detalhes sobre o método de aproximação por amostragem, incluindo avaliação da qualidade das aproximações, o leitor pode recorrer a [Bianchi \[2003\]](#).

5.4 Otimização por Colônia de Formigas

Apesar de existirem outras metaheurísticas que podem ser aplicadas ao problema do PTSP, a ACO tem se mostrado a metaheurística com melhor desempenho e, como o algoritmo de estado da arte atual é baseado nela, ela será a heurística apresentada neste documento.

A inspiração da ACO vem do comportamento de formigas reais, descrito em [Deneubourg et al. \[1990\]](#), que permite que elas encontrem menores caminhos entre seus ninhos e as fontes de alimentos. Enquanto as formigas andam de seus ninhos para as fontes de alimentos e vice-versa, elas depositam uma substância chamada feromônio no chão. Quando elas decidem por um caminho a seguir, os caminhos marcados por maiores concentrações de feromônios tenderão a serem seguidos. O algoritmo 5 exhibe o funcionamento geral dessa metaheurística.

Algoritmo 5 Otimização por colônia de formigas genérico

```

while Condição de parada não atendida do
    atividade_das_formigas()
    evaporacao_de_feromonios()
    outras_acoes()
end while

```

Numa abordagem geral de um algoritmo de ACO para o PTSP, a cada iteração cada uma das m formigas constroem um *tour* através de todas as n cidades da instância. Partindo de uma cidade aleatória, uma formiga procede escolhendo continuamente a próxima cidade a partir de sua posição atual até que o *tour* é completado ao retornar da última cidade para a primeira. Na construção do *tour*, a seleção de uma cidade é dada baseada em

uma informação heurística e numa concentração de feromônio. A informação heurística representa quão bom parece ser ir da cidade i para a cidade j . Os valores de feromônios servem como uma forma de memória, adaptada através do tempo para indicar quais escolhas foram boas num passado próximo. Um método para escolha da próxima cidade pode ser expresso pela equação 23⁸, onde τ_{ij} representa os valores de feromônios, η_{ij} representa o conhecimento heurístico, S é o conjunto de escolhas disponíveis e α e β são constantes que determinam a influência relativa dos valores heurísticos e valores de feromônios nas decisões das formigas.

$$p_{ij} = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta}}{\sum_{h \in S} \tau_{ih}^{\alpha} \eta_{ih}^{\beta}} \quad (23)$$

Após cada uma das m formigas construírem uma solução, a informação de feromônios é atualizada. Primeiro, uma parcela dos antigos feromônios é evaporada e, depois, uma atualização positiva é realizada sobre os melhores *tours* encontrados.

Além das atividades das formigas e das atualizações de feromônios, outras ações podem ser realizadas, como a ativação de um procedimento de otimização local, ou a coleta de informações globais que podem ser usadas para decidir se é útil ou não depositar mais ou menos feromônios em uma determinada região, um exemplo deste caso é para depositar mais feromônios no caminho da formiga que encontrou o menor caminho.

Na ACO, as heurísticas de estimação da esperança do tamanho do caminho apresentadas na seção anterior podem ser usadas como estimadores para a colônia de formigas.

5.5 Estado da arte – pACS+2.5-opt-EEais

Esta seção apresenta o pACS+2.5-opt-EEais, uma combinação de algoritmos publicados em Balaprakash et al. [2008], que, até o presente momento, é tida como o estado da arte para solução heurística do PTSP.

O pACS+2.5-opt-EEais é baseado no algoritmo pACS+1-shift. O pACS+1-shift, por sua vez, é um algoritmo de ACO padrão onde, a cada iteração, m formigas constroem soluções da seguinte forma: com uma probabilidade q_0 , uma formiga k no nó i escolhe mover para o nó j que maximiza o produto $\tau_{ij} \eta_{ij}^{\beta}$; com probabilidade $1 - q_0$, o próximo nó j é escolhido com probabilidade descrita na equação 23.

Além da função heurística da ACO supracitada, chamada pACS, a busca local 1-shift é aplicada a todas as soluções obtidas pelas formigas antes da etapa de atualização de feromônios. O algoritmo procede em duas fases: a primeira consiste em explorar a vizinhança das soluções, onde o conjunto de vizinhos de uma dada solução contém todas as soluções que podem ser obtidas trocando dois nós consecutivos. A segunda fase explora a vizinhança de inserção de nós em ordem lexicográfica. As diferenças de custos são obtidas por avaliação delta, uma técnica que considera apenas a contribuição de custo de componentes de soluções que não são comuns entre duas soluções.

O 2.5-opt-EEais, por sua vez, difere do 1-shift aplicado ao pACS nos seguintes elementos:

1. Adota estimação empírica, confira a seção 5.3, na avaliação delta.

⁸Apresentada em Branke and Guntch [2004]

2. Se baseia na heurística 2.5-opt do TSP, apresentada em Bentley [1992];
3. Explora técnicas de redução de vizinhança do TSP.

Além disso, sua efetividade é aumentada pelo uso de técnicas de redução de variância. Para detalhes sobre sua implementação, confira Balaprakash et al. [2008] e Balaprakash et al. [2007b].

6 Exercícios Propostos

1. Calcule a diferença entre os *tours* $\tau = 1, 2, 3, 4, 5, 6$ e $\tilde{\tau} = 1, 5, 4, 3, 2, 6$ com os vértices sobre os lados de um hexágono regular de lado 1 colocado sobre o plano euclidiano, com cada vértice com probabilidade $p = 0.5$ usando a heurística de busca local 2-p-opt.
2. Calcule a diferença entre os *tours* $\tau = 1, 2, 3, 4, 5, 6$ e $\tilde{\tau} = 1, 3, 4, 5, 2, 6$ com os vértices sobre os lados de um hexágono regular de lado 1 colocado sobre o plano euclidiano, com cada vértice com probabilidade $p = 0.5$ usando a heurística de busca local 1-shift.
3. Sempre é melhor usar um método de estimação do valor da função objetivo? Justifique.
4. Proponha uma aplicação do PTSP em localização de facilidades.
5. Para quais casos se pode ter a esperança de se resolver o PTSP de maneira exata? Esse método de solução é escalável?
6. Qual o maior problema das formulações lineares do PTSP?

Referências bibliográficas

- P. Balaprakash, M. Birattari, and T. Stützle. Engineering stochastic local search algorithms: A case study in estimation-based local search for the probabilistic traveling salesman problem. Technical report, IRIDIA, 2007a.
- P. Balaprakash, M. Birattari, T. Stützle, and M. Dorigo. Adaptive sample size and importance sampling in estimation-based local search for the probabilistic traveling salesman problem: A complete analysis. Technical report, IRIDIA, Université Libre de Bruxelles, 2007b.
- P. Balaprakash, M. Birattari, T. Stützle, Z. Yuan, and M. Dorigo. Estimation-based ant colony optimization and local search for the probabilistic traveling salesman problem. Technical report, IRIDIA, 2008.
- J. J. Bartholdi and L. K. Platzman. An $(n \log n)$ planar traveling salesman heuristic based on spacefilling curves. *Operations Research Letters*, 1:121–125, 1982.
- J. J. Bartholdi, L. K. Platzman, R. L. Collins, and W. H. Warden. A minimal technology routing system for meals on wheels. *Interfaces*, 13:1–8, 1983.
- J. L. Bentley. Fast algorithms for geometric traveling salesman problems. *ORSA Journal of Computing*, 4:387–411, 1992.
- D. J. Bertsimas. *Probabilistic combinatorial optimization problems*. PhD thesis, Massachusetts Institute of Technology, 1988.
- D. J. Bertsimas and L. H. Howell. Further results on the probabilistic traveling salesman problem. *European Journal of Operational Research*, 65:68–95, 1993.
- L. Bianchi. *New approaches for solving the Probabilistic Traveling Salesman Problem*. PhD thesis, Université Libre de Bruxelles, IRIDIA, 2003.
- L. Bianchi and J. Knowles. Local search for the probabilistic traveling salesman problem: A proof of the incorrectness of bertsimas' proposed 2-p-opt and 1-shift algorithms. Technical report, IRIDIA, 2002.
- L. Bianchi, L. M. Gambardella, and M. Dorigo. An ant colony optimization approach to the probabilistic traveling salesman problem. In *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, volume 2439, pages 883–892, 2002a.
- L. Bianchi, L. M. Gambardella, and M. Dorigo. Solving the homogeneous probabilistic travelling salesman problem by the aco metaheuristic. In *Ant Algorithms, 3rd International Workshop, ANTS 2002, volume 2463 of LNCS*, pages 176–187. Springer, 2002b.
- L. Bianchi, J. Knowles, and N. Bowler. Local search for the probabilistic traveling salesman problem: correction to the 2-p-opt and 1-shift algorithms. *European Journal of Operational Research*, 162:206–219, 2005.
- J. Borenstein, H. R. Everett, and L. Feng. Where am i? sensors and methods for mobile robot positioning. Technical report, University of Michigan, 1996.

- N. E. Bowler, T. M. A. Fink, and R. C. Ball. Characterization of the probabilistic traveling salesman problem. *Physical review.*, 68:036703.1–036703.7, 2002.
- J. Branke and M. Guntsch. New ideas for applying ant colony optimization to the probabilistic tsp. *Lecture Notes in Computer Science*, 2611:127–134, 2003.
- J. Branke and M. Guntsch. Solving the probabilistic tsp with ant colony optimization. *Journal of Mathematical Modelling and Algorithms*, 3:49300425, 2004.
- J. L. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behaviour*, 3:159–168, 1990.
- M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1: 53–66, 1997.
- M. Flood. The traveling salesman problem. *Operations Research*, 4:61–75, 1956.
- P. Jaillet. *Probabilistic Traveling Salesman Problems*. PhD thesis, Massachusetts Institute of Technology, 1985.
- A. Jézéquel. Probabilistic vehicle routing problems. Master’s thesis, Massachusetts Institute of Technology, 1985.
- G. Laporte, F. V. Louveaux, and H. Mercure. A priori optimization of the probabilistic traveling salesman problem. *Operations Research*, 42:543–549, 1994.
- C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization : Algorithms and Complexity*. Dover Publications, July 1998. ISBN 0486402584.
- F. A. Rossi and I. Gavioli. Aspects of heuristic methods in the probabilistic traveling salesman problem. In *Stochastics in Combinatorial Optimization*, 1987.
- R. Siegwart and I. R. Nourbakhsh. *Introduction to Autonomous Mobile Robotics*. MIT Press, 2004.

A Implementação e experimentação

Este capítulo descreve o trabalho de implementação associado a este minicurso, incluído como apêndice por não se encaixar de melhor forma em outro lugar deste documento. Este capítulo apresenta um problema hipotético a ser resolvido com o uso do PTSP e descreve possíveis abordagens para este problema.

Considere o problema hipotético de um robô cuja função é desarmar o maior número de minas possível em um campo minado onde as posições das minas são conhecidas a priori, mas para sua localização o robô se baseia somente na odometria⁹. Com este cenário hipotético, é possível aplicar um modelo de erros de odometria ao mapa contendo as posições das minas para estimar a probabilidade com que o robô as alcançará. Assuma, sem perda de generalidade, que a mera passagem sobre a posição de uma mina garante o seu desarmamento.

A.1 Metodologia

A seguinte metodologia foi adotada para implementação e experimentação:

1. Implementação do modelo matemático no *solver* iLOG CPLEX e resolução de problemas pequenos e comparação dos resultados com a heurística de estado da arte atual, pACS+2.5-opt-EEais;
2. Execução da heurística pACS+2.5-opt-EEais a uma instância do PTSP gerada de um problema sintético usando o modelo de erro de odometria de um robô, descrito na sessão Modelo de erro de odometria;
3. Aplicar simulação para comparação de resultados;
4. Analisar os resultados obtidos e descrever se sua utilização é viável para o problema proposto e se as soluções obtidas correspondem à idéia intuitiva, de que pontos com menor probabilidade estariam em estágios finais do *tour*;

Devido à complexidade dos modelos matemáticos e a curva de aprendizado necessária para utilizar o *solver* iLOG CPLEX, os testes descritos no item 1 não foram realizados.

Modelo de erro de odometria

Existem diversos trabalhos na literatura que tratam de erros de odometria, por não fazer parte do escopo deste documento discutir sobre esse assunto de maneira exaustiva, leitores interessados podem recorrer a [Siegwart and Nourbakhsh \[2004\]](#) e [Borenstein et al. \[1996\]](#). Para estimação do erro de odometria, foi desenvolvida a heurística descrita no algoritmo 6. Sua idéia básica é gerar uma função gaussiana que indica a probabilidade de o robô alcançar determinado ponto a partir da distância percorrida pelo robô a partir do primeiro ponto do grafo.

⁹A localização por odometria consiste em ler dados do odômetro que, grosso modo, converte as rotações das rodas do robô na distância percorrida – dado que se conhece o raio das rodas e o número de rotações das rodas, é possível estimar a distância percorrida, note que, devido a irregularidades de terreno, esse método de medição inerentemente possui erros.

Algoritmo 6 Heurística de aproximação de erro de odometria

Entrada: I, F

$$\text{dist} \leftarrow \sqrt{(I_x - F_x)^2 + (I_y - F_y)^2}$$

$$\mu_x \leftarrow \text{dist} \cdot \psi_x$$

$$\mu_y \leftarrow \text{dist} \cdot \psi_y$$

$$\sigma \leftarrow \sqrt{\mu_x \mu_y \eta}$$

$$p \leftarrow \text{gauss}(0, \sigma)$$

return p

Nesse algoritmo, $\text{gauss}(\mu, \sigma)$ corresponde à função gaussiana de probabilidades com média μ e desvio padrão σ , conforme descrito pela equação 24 e ϕ_x, ϕ_y e η são constantes para regulagem da probabilidade retornada.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (24)$$

A.2 Experimentos

Para execução desse trabalho foram realizados os experimentos já citados na metodologia, excluída a implementação em método exato devido à dificuldade de implementação do problema escolhido.

A tabela 2 exibe os resultados obtidos da seguinte forma:

1. Instâncias do TSP no formato TSPLIB foram geradas com auxílio da ferramenta de geração de instâncias DIMACS, que foi alterada para gerar pontos em um quadrado de lado 200, pois, para quadrados maiores, a probabilidade calculada pelo modelo de erro usado era muito baixa para os nós, implicando em baixa probabilidade de visitas;
2. Foi aplicado o algoritmo 6 às instâncias geradas pelo DIMACS, convertendo-as a instâncias heterogêneas do PTSP;
3. O algoritmo pACS+2.5-opt-EEais foi aplicado a essas novas instâncias para sua solução;
4. Foi realizada uma “simulação” sobre quais nós seriam incluídos em um tour real, para cada instância foram coletadas trezentas amostras de instâncias do PTSP e essas amostras foram resolvidas usando a heurística DMST para solução do TSP¹⁰.

A.3 Conclusões e trabalhos futuros

Deste trabalho foi possível perceber a complexidade do PTSP e identificar uma aplicação nova, consistindo de um cenário hipotético de um robô desarmador de minas. Para trabalhos futuros, a aplicabilidade desta abordagem deve ser mais profundamente estudada com simulações mais fiéis e melhores modelos de erros.

¹⁰Os nós sorteados passavam a fazer parte permanente do tour enquanto os nós que não foram sorteados deixavam de fazer parte do tour, transformando o problema em um TSP

nós	pACS+2.5-opt-EEais		Simulação	
	tempo(s)	custo	tempo(s)	custo
10	< 0.000010	347	—	372
20	0.000010	465	—	561
50	0.000010	916	—	1085
100	0.004010	1249	—	1584
200	0.020011	1749	—	2315
300	0.048013	2234	—	2787
500	0.124018	2846	—	3869
600	0.180021	3165	—	3976
700	0.244025	3073	—	4196
1000	0.488040	4011	—	5186
2000	1.992134	5695	—	7894
3000	4.568295	6859	—	9481
9999	69.49235	—	—	17043

Tabela 2: Tabela exibindo tempo de execução e custo de soluções dos algoritmos pACS+2.5-opt-EEais para o PTSP e DMST para o TSP “simulado”. Os campos marcados com — não tiveram seu tempo obtido e, no caso do custo da solução de 9999 nós para PTSP, não foi possível computar o custo da solução em tempo hábil.

B Soluções dos exercícios propostos

1. $|E[L(\tilde{\tau})] - E[L(\tau)]| = |3.967548 - 4.144431| = 0.176883$.¹¹
2. $|E[L(\tilde{\tau})] - E[L(\tau)]| = |4.285056 - 3.967548| = 0.317508$.¹²
3. Claramente, a resposta a esta questão é não, principalmente por sua generalidade. É necessário ponderar sobre o tempo de obtenção de uma aproximação que seja “boa o suficiente”, que é dependente do problema e das exigências do proponente do problema. Se o erro aceitável for mínimo ou se a estrutura do problema PTSP não satisfizer certas condições, a aplicação de uma aproximação pode não ajudar na solução do PTSP, como é característico de aproximações.
4. A abordagem tradicional de localização de facilidades assume que cada vez que um cliente faz uma requisição, ele será servido exclusivamente por um veículo que visita este consumidor e que, depois, volta imediatamente para a fábrica; Em outras palavras, minimiza-se a distância média de ou para uma facilidade para a população de usuários considerando uma média ponderada das distâncias entre as facilidades e os clientes. No entanto, é possível encontrar soluções práticas onde veículos servirão a mais de um cliente por viagem. Este problema já foi proposto como um TSP através de todos os subconjuntos possíveis dos n clientes. A localização ótima, neste caso, a cada passo necessitava de $2^n - 1$ TSP para serem resolvidos. Ao invés disso, é possível ver este problema como uma alternativa de projetar um tour a priori através dos n clientes assumindo uma metodologia do PTSP, ou seja, localizar a facilidade

¹¹Exercício retirado do Apêndice B de Bianchi [2003]

¹²Exercício retirado do Apêndice C de Bianchi [2003]

de modo a minimizar o comprimento esperado de um tour a priori através de todos os consumidores.

5. Normalmente, em casos que se assemelhem ao TSP, ou seja, com a maioria dos nós com probabilidade de visita alta. Outros casos são casos muito pequenos, onde até por enumeração completa pode ser possível resolver o PTSP. Claramente, essa solução não é escalável devido à explosão combinatória, pois o PTSP é ainda mais difícil que o TSP, que já sofre com esse problema.
6. O maior problema das formulações lineares do PTSP apresentadas neste documento são o número de restrições e o número de variáveis de decisão, que tornam o problema extremamente grande e difícil de resolver. Ainda assim, falta na literatura uma documentação recente sobre quantos nós podem ser resolvidos por esses métodos, já que a maioria das soluções recentes utilizam soluções heurísticas para o PTSP.