

RESOLVENDO O PROBLEMA DO CAIXEIRO VIAJANTE VIA PROCEDIMENTO DE METAHEURÍSTICA GRASP

Filipe Herculano Rocha
Vicente Ferreira Oliveira
Lucas Remígio de Sousa Bezerra
Gabriel Angelo Freire Gonçalves

Universidade Estadual do Ceará
Departamento de ciência da computação

20 de Novembro de 2018

Resumo – A maioria dos problemas de otimização combinatória pertencem à classe NP, o que significa que sua solução ótima por meio de métodos enumerativos pode exigir um tempo de processamento inviável. Assim, o uso de heurísticas é frequentemente mais interessante, pois são capazes de encontrar uma relação custo/benefício aceitável entre a qualidade da solução e o tempo de processamento. Este artigo propõe uma meta-heurística para solução do problema do caixeiro viajante (TSP), em que um procedimento de busca adaptativa aleatória gulosa (GRASP) é implementado.

Palavras-chave – Problema do caixeiro viajante, GRASP, otimização combinatória;

1 Introdução

O estudo de problemas de otimização combinatória é relevante tanto do ponto de vista teórico quanto prático. Estes problemas são comumente encontrados em diversas situações reais, tais como, roteamento de veículos (“Vehicle Routing Problems” (VRP)) e programação de tabela de horários (“Scheduling”), e são usualmente classificados como NP-completo ou NP-difícil (Lewis and Papadimitriou, 1997), o que implica na não existência de uma solução determinística capaz de ser executada em tempo polinomial. Por isso, o uso de heurísticas é frequentemente mais interessante (Glover and Kochenberger, 2003), pois são capazes de encontrar uma relação custo/benefício aceitável entre a qualidade da solução e o tempo necessário para o seu cálculo.

2 Definição do problema

O Problema do Caixeiro Viajante (TSP, do inglês Traveling-Salesman Problem) é um problema NP-Difícil da área de otimização que possui inúmeras aplicações práticas. O TSP modela diversas situações reais, como o roteamento de veículos para atender chamados ou

ocorrências. Por pertencer à classe dos problemas NP-Difíceis, não há algoritmos eficientes para resolvê-lo na exatidão. O TSP também é encontrado como subproblema de modelagens maiores, que envolvem muitas vezes vários problemas pertencentes às classes NP-Completa e NP-Difícil, como o Problema da Mochila (Martello e Toth, 1990).

O TSP consiste em, dado um grafo completo $G(V, E)$, com n vértices, obter um ciclo hamiltoniano de custo mínimo, isto é, deseja-se, a partir de um vértice inicial, passar por todos os demais vértices do grafo uma única vez e então retornar ao vértice inicial. Cada aresta que liga um par de vértices do grafo possui um custo $c(i, j)$ que determina o quanto se gasta para ir de i até j . Esse custo pode ter diversos significados, de acordo com a aplicação desejada. Ele pode representar, por exemplo, a distância, o tempo ou mesmo o preço para se deslocar entre duas cidades i e j . Considerando esses custos, o objetivo do problema é obter um ciclo hamiltoniano de custo total mínimo a partir de um dado ponto inicial. Esse custo total é dado pela soma dos pesos de todas as arestas contidas no ciclo.

A definição de um ciclo hamiltoniano é dada da seguinte forma: Dado um grafo $G(V, E)$ com n vértices, deseja-se obter um ciclo (v_1, \dots, v_n) que parta de um vértice inicial v_1 , passe por todos os demais $n-1$ vértices de V uma única vez e retorne a v_1 . A definição formal pode ser escrita como se segue: Dados um grafo completo $G(V, E)$ e uma função custo $c : V \times V \rightarrow \mathbb{Q}^+$, encontrar uma permutação (v_1, v_2, \dots, v_n) de V tal que

$$c(v_n, v_1) + \sum_{i=1}^{n-1} c(v_i, v_{i+1})$$

seja mínimo.

3 Descrição do algoritmo e complexidade

O GRASP (Greedy Randomized Adaptive Search Procedures) é uma metaheurística constituída por duas fases, heurísticas construtivas e busca local, que por meio de um paralelo entre gulosidade e aleatoriedade consegue na sua fase de construção gerar soluções que escapam de ótimos locais após o procedimento de busca local. Temos como vantagem que o algoritmo é pouco dependente de parâmetros, possui uma implementação relativamente simples devido ao conceito de construção gulosa aleatorizada, e é facilmente paralelizável. Além disso, o algoritmo armazena a melhor ou as melhores soluções encontradas. Hart e Shogan (1987) propuseram um mecanismo baseado em cardinalidade, no qual os k melhores candidatos são adicionados numa lista de candidatos restrita (LCR), e um mecanismo baseado em valor, no qual todas as soluções que tiverem valores melhores do que α^* melhor valor, devem compor a LCR, em que α é um valor entre 0 e 1, com $\alpha = 0$ gerando soluções puramente gulosas, enquanto que $\alpha = 1$ produz soluções totalmente aleatórias. A busca é dita adaptativa, pois a função para avaliação da gulosidade é atualizada a cada iteração da heurística construtiva. Para a calibração do parâmetro α , em geral as seguintes estratégias podem ser utilizadas:

- Uso de valor fixo;
- Uso de valor escolhido aleatoriamente em uma distribuição uniforme;
- Uso de valor escolhido aleatoriamente em uma distribuição empírica;
- Estratégia reativa.

Fixar o valor de α , em geral, leva a resultados inferiores, pois o GRASP retarda o processo de obtenção de soluções superiores que poderiam vir a ser obtidas com outro valor de α . Já com a estratégia reativa consiste em determinar o valor de α de acordo com a distribuição de

probabilidades que vai ser gerada ao longo das iterações do algoritmo, e é atualizada de forma dinâmica e seletiva.

4 Resultados dos experimentos computacionais

Instâncias			Solução "Ótima"	Resultados Obtidos			
N.	Referências	Cidades		Inicial	Tempo	Melhor	Aproxim ação
1	gr17.tsp	17	2085	2187	1.711 ms	2085	0%
2	gr24.tsp	24	1272	1553	2.876 ms	1272	13.76%
3	gr48.tsp	48	5046	6098	11.642 ms	5917	17.26%
4	si175.tsp	175	21407	22263	144.222 ms	22137	3.41%
5	si535.tsp	535	48450	50144	1721.72 ms	49898	2.99%
6	si1032.tsp	1032	92650	94571	5898.96 ms	94534	2.03%

5 Conclusões

Esta é uma simples implementação da metaheurística chamada GRASP. Este projeto tenta resolver o problema do Caixeiro Viajante, dada uma série de topologias tomadas da TSPLIB. O algoritmo GRASP tem dois parâmetros principais: o número de vezes que o GRASP é executado e o valor do alfa. Simplificando, esses números são estáticos e não mudam durante a execução. (alfa = 0.85 e kMaxIter = 30).

Referências

Blum, and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison.

Lin, S. and Kernighan, B. (1973). An effective heuristic algorithm for the traveling-salesman problem, *Operations Research* 21(2): 498–516

<http://www.decom.ufop.br/prof/marcone/Orientacoes/PCVCP-Exato-VNS.pdf>

<http://www.inf.ufpr.br/aurora/disciplinas/topicosia2/downloads/trabalhos/GraspTSP.pdf>