https://eventos.utfpr.edu.br//sicite/sicite2019

Operador de cruzamento de partições aplicado ao problema da árvore de Steiner em grafos

Partition crossover operator applied to Steiner tree problem in graphs

Bruna Almeida Osti
brunaosti@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, PR, Brasil

Danilo Sipoli Sanches
danilosanches@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, PR, Brasil

RESUMO

Neste trabalho, será investigado a eficiência do algoritmo genético para otimizar as soluções para o Problema da Árvore de Steiner em Grafos (STPG), sendo aplicado o operador de Cruzamento de Partição Generalizado (GPX), para caminhar entre os mínimos locais, buscando analisar a eficiência, qualidade e tempo de execução do operador comparado a outros algoritmos de otimização do problema. Em geral, o operador de cruzamento de partição generalizado tem como princípio aproveitar as melhores partes de duas soluções, garantindo sempre que a melhor solução gerada seja sempre melhor ou que mantenha o custo das soluções iniciais, sem aumentar a complexidade computacional do operador. Os dados e as análises mostraram que o GPX traz melhorias para as soluções do STPG, mas acima disso mostraram que o modelo é funcional, portanto, é possível reutilizá-lo para outros problemas de otimização combinatória em grafos, com outros tipos de restrições apenas alterando algumas estruturas do algoritmo.

PALAVRAS-CHAVE: Algoritmos Genéticos. Otimização Combinatória. Teoria dos grafos.

Recebido: 19 ago. 2019.

Aprovado: 01 out. 2019.

ABSTRACT

In this work, the efficiency of the genetic algorithm to optimize the solutions for the Steiner Tree Problem in Graph (STPG) will be investigated, applying the Generalized Partition Crossover (GPX) operator to walk between the local minimums, seeking to analyze the operator efficiency, quality and runtime compared to other problem optimization algorithms. In general, the generalized partition crossover operator has the principle of taking advantage of the best parts of two solutions, always ensuring that the best generated solution is better or that it keeps the cost of the initial solutions without increasing the computational complexity of the operator. The data and analysis showed that GPX brings improvements to STPG solutions, but above that showed that the model is functional, so you can reuse it for other combinatorial optimization problems in graphs, with other constraints just changing some structures of the algorithm.

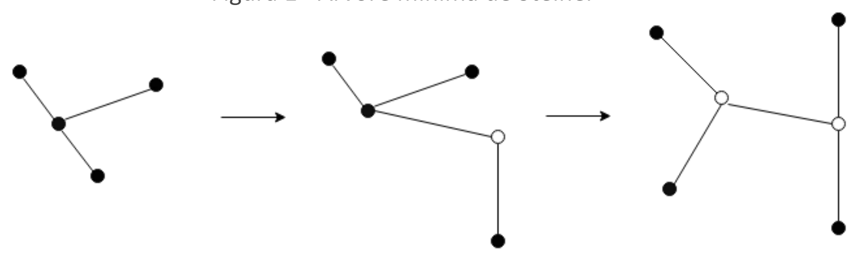
KEYWORDS: Genetic Algorithm. Optimization. Theory of graphs.

INTRODUÇÃO

A otimização combinatória é constantemente utilizada para modelar e resolver grandes problemas de logística envolvendo variáveis discretas, tendo como exemplo um dos problemas mais conhecidos e estudados nessa área: a árvore de Steiner em grafos (STPG), sendo um problema de complexidade NP-Difícil, mas de ampla eficácia em projetos de circuitos elétricos, redes de comunicação e de transporte (COELHO, 2016).

Sobretudo, um grafo é um conjunto interligado de arestas e vértices, no caso cada aresta está associada a dois vértices. Considerando um grafo não direcionado $G = (V, A)$, na qual **V** são os vértices, e **A** são as arestas, com pesos associados às arestas (KAPSALIS et al., 1993). O problema da árvore de Steiner consiste em encontrar um subgrafo de G que percorra todos os vértices terminais e que contenha o menor peso total de arestas, formando-se uma árvore mínima, como mostra a Figura 1, na qual os vértices terminais são representados pelos pontos pretos e os não-terminais pelos pontos brancos.

Figura 1 - Árvore mínima de Steiner



Fonte: Autoria Própria (2019).

Os vértices terminais são chamados de pontos obrigatórios e todos os outros vértices do conjunto são opcionais, ou seja, os vértices opcionais que entram na solução são chamados de pontos de Steiner, portanto, a solução final abrange apenas um subconjunto que é denominada árvore de Steiner.

Neste trabalho, será investigado a eficiência do algoritmo genético para otimizar as soluções para o problema da árvore de Steiner em grafos, sendo aplicado o operador de cruzamento de partição generalizado (GPX), para caminhar entre os mínimos locais, baseado no operador de cruzamento de partição (PX) proposto por WHITLEY et al. (2009). O operador GPX apresentou uma boa performance para encontrar soluções, quando aplicado sobre o problema para o qual foi primeiramente definido, o problema do caixeiro viajante (*Travelling salesman problem, TSP*) (WHITLEY et al., 2010). De mesmo modo, será realizado a comparação com o algoritmo proposto por KAPSALIS et al. (1993), além do algoritmo mais eficiente para encontrar árvores mínimas (PRIM, R., 1957). Nos testes será levado em conta o tempo, custo das soluções e qualidade das soluções.

MATERIAIS E MÉTODOS

Um dos problemas mais comuns na implementação da solução para o problema da árvore de Steiner é a necessidade do poder computacional envolvido no processamento, pois todo o processo acaba sendo muito custoso devido a quantidade de iterações necessárias e o tamanho dos dados disponíveis (COELHO, 2016). Deste modo, a utilização de um método heurístico que trouxesse um resultado aproximado mas muito preciso acaba tornando-se indispensável, neste caso optamos pela utilização de algoritmos genéticos em conjunto com o operador de cruzamento de partição generalizado (GPX).

O GPX tem como princípio aproveitar as melhores partes de duas soluções, garantindo sempre que a melhor solução gerada seja melhor ou que mantenha o custo das soluções iniciais. Primeiramente, o PX (cruzamento de partição) utiliza a união de duas ou mais soluções para construir um grafo e tem um tipo específico de partição no grafo construído. Se tal partição for encontrada, duas soluções são construídas. Por outro lado, o cruzamento de partição generalizada (GPX) é uma versão generalizada do cruzamento de partição (PX). Pois, na maioria dos casos, o grafo de união construído de dois pais contém várias partições do tipo específico que o cruzamento de partição está procurando, portanto, o cruzamento generalizado de partições é capaz de encontrar e usar todas essas partições em uma única recombinação sem aumentar a complexidade computacional do operador (WHITLEY et al., 2010).

Em geral, o operador de cruzamento de partições generalizado gera um grafo de união G = (V, E) através de dois grafos de entrada e retira os vértices em comum nas duas soluções, tornando-se um conjunto de partições desconexas, portanto utilizamos a busca em largura (BFS) para achar cada partição contida no subgrafo, como mostra o Algoritmo 1 (TINÓS, 2019).

Algoritmo 1 - GPX para o problema da árvore de Steiner

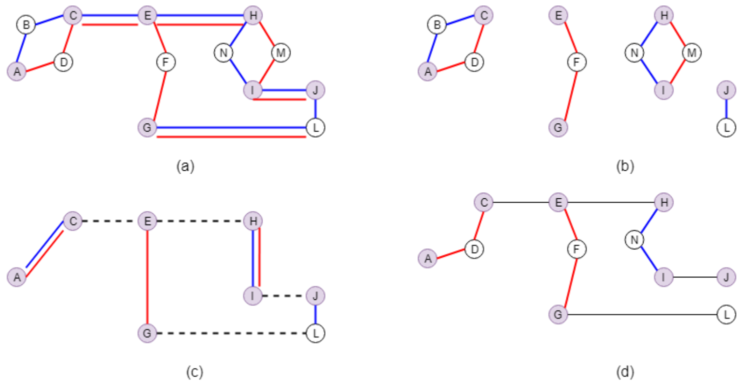
Entrada: G1, G2

Saída: Combinação de dois grafos de entrada

a) Crie um grafo de união $G_u = G_1 \cup G_2$ através dos grafos G_1 e G_2 (pais);
b) Remova todas as arestas em comum de G_u;
c) Encontre todos os componentes conexos em G_u. Os componentes conectados com mais de um nó são candidatos a recombinação;
d) Para cada componente candidato, crie 2 grafos simplificados, um para cada parente. Os grafos simplificados contém apenas os vértices de entrada do componente candidato, vértices conectados com pelo menos uma aresta entre vértices localizados em diferentes componentes candidatos. As arestas dos grafos simplificados podem ser obtidas removendo os vértices que não são vértices de entrada (e ligando os vértices vizinhos);
e) Teste cada componente candidato. Se dois grafos simplificados são iguais, então o componente candidato é um componente recombinante;
f) O grafo considerado também é considerado um componente recombinante;
g) Aplique o operador de cruzamento selecionando a menor solução parcial dentro de cada componente recombinante.

A Figura 2 descreve o processo de execução do algoritmo GPX, além disso descreve o processo de seleção das partições como componentes recombinantes para uma nova solução e estabelece o que é considerado um grafo simplificado.

Figura 2 - Aplicação do GPX com componente não recombinante



Fonte: TINÓS, R. (2019).

No qual,

a) Grafo de União formado pelos pais vermelho e azul;

b) Depois de removido as arestas comuns, quatro componentes candidatos são encontrados;

c) Os grafos simplificados são criados retirando-se os vértices internos, e vértices que não estão conectados a vértices em outros componentes candidatos. As arestas em comum são representadas pelas linhas tracejadas; Os grafos simplificados para dois componentes (formado pelos vértices A e C, e pelos vértices H e I) são respectivamentes iguais, portanto, eles são componentes recombinantes. Os dois outros componentes candidatos não são componentes recombinantes porque seus grafos respectivos não são iguais. Depois de identificar os componentes recombinantes isolados, o resultado do grafo também é considerado um componente recombinante. Portanto, há 3 componentes recombinantes neste exemplo: componente 1 (Formado pelos vértices: A, B, C e D), 2 (Formado pelos vértices H, I, M e N), e o 3 (Formado pelos vértices E, F, G, J, e L);

d) A solução formada pela seleção: vermelho pelo componente 1; azul pelo componente 2, e vermelho pelo componente 3.

Para os testes, Foram escolhidos dois datasets para a árvore de Steiner em grafos contidos na OR-LIBRARY que é mantida por JE Beasley pesquisador de otimização combinatória pela Brunel University – Londres (BEASLEY, 1990). Portanto, escolheu-se os datasets do conjunto B, por oferecerem uma melhor visualização do conjunto, no qual os dois datasets escolhidos dentro do conjunto B foram: **B01**, **B07**.

RESULTADOS E DISCUSSÕES

Com a realização dos testes foi possível obter diversos dados sobre o comportamento e desempenho dos métodos STPG. A seguir serão demonstrados tabelas apresentando informações e discussões sobre os testes, na qual cada configuração foi executada dez vezes e calculou-se a média e o desvio padrão da amostra em análise. Cada tabela está mostrando uma base de dados (*dataset*) e para cada uma são comparados a melhor solução encontrada, a solução média encontrada, além do tempo médio de processamento do algoritmo. Os métodos analisados estão listados na seguinte ordem: GPX, Kapsalis, MST e o ótimo concedido pelo *dataset*.

A Tabela 1 apresenta os resultados dos testes com o *dataset B01*. Nestes testes foi possível perceber que o GPX foi superior a todos os outros métodos, apesar de nenhum se aproximar do ótimo concedido pelo dataset. De mesmo modo, a variância dos resultados manteve-se menor do que nos outros métodos.

Tabela 1 - Comparação dos métodos para B01

Operador	Melhor Solução Encontrada	Solução Média	Tempo Médio (s)
GPX	182	204,0 (±16,1)	0,1265 (±0,0033)
Kapsalis	186	220,8 (±21,5)	0,3208 (±0,1245)
MST	238	-	-
Ótimo	82	-	-

Fonte: Autoria própria (2019).

A Tabela 2 apresenta os resultados dos testes com o *dataset B07*. Nestes testes foi possível perceber que o GPX foi superior a todos os outros métodos, e a sua melhor solução encontrada se aproximou do ótimo concedido pelo *dataset*. Além disso, sua solução média diminuiu consideravelmente o custo em comparação com os outros métodos.

Tabela 2 - Comparação dos métodos para B07

Operador	Melhor Solução Encontrada	Solução Média	Tempo Médio (s)
GPX	257	301,5 (±25,2)	0,0876 (±0,0342)
Kapsalis	320	331,5 (±07,7)	0,5464 (±0,0611)
MST	341	-	-
Ótimo	238	-	-

Fonte: Autoria própria (2019).

Os testes apresentados permitiram ter uma boa análise dos métodos, além disso, o GPX mostrou-se superior nas aplicações testadas, pois aproximou-se do ótimo global do dataset e manteve-se com baixa variância de dados.

O tempo dos testes para o GPX é menor do que para o algoritmo baseado no Kapsalis, mas deve-se levar em conta que as entradas para o operador devem ser factíveis (conexas), ou seja, é necessário a utilização de outros algoritmos para a geração dos grafos de entrada.

O algoritmo obteve alguns casos no qual gerou-se árvores desconexas. Entretanto, apenas uma porcentagem de 10% de todas as amostras apresentaram esta anomalia, mantendo-se assim a qualidade nas soluções geradas.

CONCLUSÃO

Os dados e as análises mostraram a melhoria que o GPX traz para as soluções do STPG, mas acima disso mostraram que o modelo é funcional, e que através de futuras adaptações o algoritmo se tornará mais rápido e preciso. Pois, nos dois *datasets* escolhidos o GPX obteve resultados superiores aos demais em quesitos de tempo, proximidade com o ótimo do *dataset* e qualidade da solução, o que mostra que o operador consegue desenvolver e otimizar as soluções com melhor aproveitamento. Embora diversos problemas tenham sido enfrentados durante o desenvolvimento, como encontrar meios de garantir que as soluções geradas fossem factíveis (conexas) a gama de possibilidades do uso do GPX é muito grande, é possível reutilizá-lo para outros problemas de otimização combinatória em grafos, com outros tipos de restrições apenas alterando algumas estruturas do algoritmo.

Em trabalhos futuros se estudará a possibilidade de melhorias quanto a qualidade das soluções e otimização de tempo para tornar o algoritmo melhor para *datasets* maiores, e também a possibilidade de aplicação em problemas de natureza reais.

AGRADECIMENTOS

Agradecemos à UTFPR que disponibilizou a utilização a infraestrutura acadêmica para o desenvolvimento deste trabalho. Agradecemos ao docente Renato Tinós (USP - RP) que participou do desenvolvimento e testes realizados neste trabalho.

REFERÊNCIAS

KAPSALIS, A.; RAYWARD-SMITH, V. J.; SMITH, G. D. Solving the graphical Steiner tree problem using genetic algorithms. The Journal of the Operational Research Society, JSTOR, v. 44, n. 4, p. 397, abr. 1993.

WHITLEY, D.; HAINS, D.; HOWE, A. Tunneling between optima: Partition crossover for the traveling salesman problem. p. 915–922, 01 2009.

WHITLEY, D.; HAINS, D.; HOWE, A. A hybrid genetic algorithm for the traveling salesman problem using generalized partition crossover. In: Parallel Problem Solving from Nature, PPSN XI. Springer Berlin Heidelberg, 2010. p. 566–575.

PRIM, R. C. Shortest connection networks and some generalizations. Bell System Technical Journal, Institute of Electrical and Electronics Engineers (IEEE), v. 36, n. 6, p. 1389–1401, nov. 1957.

TINÓS, R. An Efficient Recombination Operator for Solving the Graphical Steiner Tree Problem. Universidade de São Paulo. Ribeirão Preto, 2019.

COELHO, J. C. O estudo das árvores de Steiner no Plano Euclidiano e algumas aplicações através do Algoritmo de Melzak. 2016. 60 f. Dissertação (Mestrado em Matemática) - Universidade Federal do Amazonas, Manaus, 2016.

BEASLEY, J.E. OR-LIBRARY, 1990. Disponível em: <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>