

SIIO, Simposio Argentino de Informática Industrial e Investigación Operativa

Evaluación computacional de los cortes locales para el problema de árbol generador con máxima cantidad de hojas

Javier Marengo^{1,2} Javier Martínez-Viademonte² Marcelo Mydlarz²

¹ Universidad de Buenos Aires

² Universidad Nacional de General Sarmiento
{jmarengo, javiermv, mmydlarz}@ungs.edu.ar

Resumen Para un problema de programación lineal entera, la técnica de *cortes locales* [1] consiste en proyectar la cápsula convexa de las soluciones factibles y una solución fraccionaria de la relajación lineal a un espacio de dimensión muy baja, encontrando ahí cortes que luego serán “elevados” al problema original; e iterar sobre este procedimiento.

La intención es obtener cortes que puedan ser aplicados en el contexto de un algoritmo de *branch-and-cut* sin recurrir a caracterizaciones previas de familias de desigualdades válidas, aprovechando fuertemente la reducción en el tamaño del problema y eligiendo una variedad de proyecciones en caso de ser conveniente.

En este trabajo estudiamos el problema de encontrar un árbol generador con máxima cantidad de hojas (MLSTP) [2] sobre un grafo conexo, un problema de interés para la industria de las telecomunicaciones [3].

Se presentan resultados y un análisis, con foco en el desempeño computacional, productos de abordar el problema con la técnica de cortes locales.

Keywords: cortes locales, branch-and-cut, programación lineal entera, MLSTP, lifting secuencial

Referencias

1. Chvátal, V. et al.: Local cuts for mixed-integer programming. *Math. Prog. Comp.* **5**, 2, 171–200 (2013). <https://doi.org/10.1007/s12532-013-0052-9>.
2. Fujie, T.: An exact algorithm for the maximum leaf spanning tree problem. *Computers & Operations Research.* **30**, 13, 1931–1944 (2003). [https://doi.org/10.1016/S0305-0548\(02\)00117-X](https://doi.org/10.1016/S0305-0548(02)00117-X).
3. Blum, J. et al.: Connected Dominating Set in Sensor Networks and MANETs. In: Du, D.-Z. and Pardalos, P.M. (eds.) *Handbook of Combinatorial Optimization*. pp. 329–369 Springer US (2004).