



HAL
open science

Combining Metaheuristics with Column Generation: Successful Approaches to Enhance Column Generation Algorithms Performance

Fabian Castaño, Marc Sevaux

► **To cite this version:**

Fabian Castaño, Marc Sevaux. Combining Metaheuristics with Column Generation: Successful Approaches to Enhance Column Generation Algorithms Performance. Sixth International Workshop on Model-based Metaheuristic (Matheuristics 2016), IRIDIA/ULB, Sep 2016, Brussels, Belgium. pp.95-100. hal-01367946

HAL Id: hal-01367946

<https://hal.science/hal-01367946>

Submitted on 30 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Combining metaheuristics with column generation: successful approaches to enhance column generation algorithms performance

Fabian Castaño¹, Marc Sevaux²

¹ Universidad Pontificia Bolivariana, Facultad de Ingeniería Industrial,
Bucaramanga, Colombia

`fabian.castano@upb.edu.co`

² Université de Bretagne-Sud, Lab-STICC, Lorient, France

`marc.sevaux@univ-ubs.fr`

Abstract. Column generation algorithms are typically adopted to address mathematical programming problems defined over a huge number of variables. This approach suffers, however, from several problems that might limit its usability. In this work some of these problems are discussed along with several strategies that take advantage of the use of (meta-)heuristics to help improve the methods performance and reduce the computational effort required to compute an optimal solution. In this work the benefits of using metaheuristic strategies within CG are discussed from the viewpoint of the way they, indirectly, address some of the causes leading to a poor performance. These different methods are tested by solving the maximum network lifetime problem in wireless sensor networks for which a model that naturally leads to column generation is considered. Experimental results show how the use of metaheuristics can generate large improvements on the performance of the basic column generation framework.

Keywords: Column generation; Metaheuristics; Matheuristics;

1 Introduction

Column generation (CG) is an efficient method used to solve large scale linear programming problems that has been largely exploited to reduce the computational effort required to solve them [1, 9, 2]. CG essentially divides a problem in two namely a restricted master problem (RMP), and a pricing subproblem (PS). RMP represents the original problem formulated over a reduced part of the solution space expressed by a subset of columns (variables). Then, the PS is used to identify additional columns that, when added to RMP, can help to further improve the objective function of RMP.

CG is an iterative method that gradually enlarges the search space by adding new variables, not considered before, that contribute to improve the objective function for RMP. RMP is initialized with a reduced set of variables that generate an initial solution for the problem and sequentially adds new variables based on the reduced cost criterion. These new unknown columns are generated at every

new iteration by using any possible method useful to solve PS, which considers the constraints that are not directly considered when solving RMP. RMP is first solved, and the dual variable values associated to the optimal solutions are used to build the objective function for PS (the reduced cost criterion). PS is then solved to check whether or not it exists a new profitable column useful to improve the objective function. If a new column is available, it is added to RMP and a new iteration is carried out; otherwise, the algorithm finishes. When the method used to solve PS can guarantee that not additional interesting columns exists, the solution to RMP is guaranteed to be optimal (at least for the linear relaxation).

Nowadays, it is widely accepted that by combining CG with metaheuristic approaches, the former method might be seriously boosted. As it could be expected, this is partially a consequence of the efficiency achieved by metaheuristics while solving the difficult optimization problems corresponding to the PS. Nonetheless, as it is shown in this work, it is also a consequence of collateral effects associated to the way metaheuristics are implemented and embedded within CG that directly tackles the causes of slow convergence. In this work, the design of metaheuristics to be embedded within CG is analyzed, and several and well known approaches used to help accelerate CG are discussed to demonstrate how CG can benefit from the characteristics and flexibility offered by metaheuristic solution approaches. Several simple ideas that can be considered when designing a CG framework for solving computationally difficult problems are evaluated through extensive computational tests that show the interest of using such approaches.

2 Metaheuristics and Column Generation

Pure CG approaches might suffer from several pathological issues that affect its performance (see for example: [13, 2]). Consequently, the use of strategies to cope with such problems is not only desirable but might be necessary. Several causes have been identified. In first place, CG demands to solve a PS at every single iteration what might cause troubles when it corresponds to a difficult optimization problem. Additionally, CG may present some issues that are inherent to the approach. One of the most typical problems corresponds to convergence, meaning that while in the first iterations the evolution of the objective function is fast, in the latter iterations these improvements may be marginal and may be reflected in a large number of iterations required to fully solve the problem (*tail-off effect*) [6, 7]. A different problem corresponds to the *heading-in effect*, which can be explained as the successive enumeration of irrelevant columns, that are unlikely to be part of the optimal solution. This phenomenon often appears throughout the first iterations of CG, while not enough information is available to produce interesting columns, and can affect heavily its performance [14].

To cope with the aforementioned problems, several strategies have been successfully applied that mostly attack the problem from the viewpoint of the dual problem [10, 12, 4]. Nonetheless, it is remarkable to see that CG obtain major benefits when it is combined with metaheuristic approaches. In the latter case, it

is typical to see that the basic structure of metaheuristics helps to improve CG performance compared to other exact approaches. Although, nowadays, MIP solvers are in a development state in which they are competitive against fast (meta-)heuristics in terms of the computational time required to solve difficult problems, the use of metaheuristics continues to be profitable as it seems they bring more reductions on the time required for solving problems when embedded in CG than its exact counterpart.

When using metaheuristics within a CG framework, three main approaches can be used to help accelerate the convergence towards the optimal solution:

- **Intensification strategies** An intensification strategy consists on returning to the RMP several interesting columns found through PS to the RMP at each iteration of CG [3, 8]. This strategy usually leads to a reduction in the number of iterations required for solving CG to achieve the optimal solution [11]. If a population based or trajectory based metaheuristic approach is used to solve PS, all the columns found during the optimization process might be saved to be added to the columns pool. The number of added columns might be limited to only κ columns such that the problem size is kept under control without losing the benefits offered by the intensification strategy.
- **Diversification strategies** Diversification strategies expand the idea of intensification. The purpose in this case is to compute at each iteration of CG an interesting set of columns contributing to different constraints [11]. This strategy might be powered if a metaheuristic approach used to solve PS is available that computes several columns simultaneously and offers naturally methods to increase diversity, *e.g.* genetic algorithms, GRASP, etc.
- **Sequential application of metaheuristics and exact approaches** In this latter case, the idea is to exploit the easiness of the process of finding new profitable columns during the first iterations of CG. In this way, it is possible to compute columns that are easy to find with not-very-strong metaheuristics that can be sequentially replaced with more sophisticated metaheuristics or exact approaches when it becomes necessary while the CG solution process evolves [2]. Finally, once the solution process evolves enough, metaheuristics might be even replaced for exact approaches so as to confirm whether or not the current solution for RMP is optimal.

3 Results and discussion

Building an efficient column generation approach to solve a difficult optimization problem is a hard task. Furthermore, solving the pricing subproblem remains a complex task that often requires developing efficient specialized approaches to face it. The methods previously mentioned are tested for solving the Maximum Network Lifetime Problem in Wireless Sensor Networks with Coverage Constraints (α -MLP) [5]. For this problem, a pretty simple model based on an exponential number of variables is available. To solve the problem, an approach based on CG is proposed for which the three aforementioned approaches are

adopted through the use of diverse metaheuristic approaches, *e.g.* Evolutionary algorithms, GRASP, VNS. As shown in Figure 1 and 2 by applying the mentioned approaches, CG can be boosted compared to its basic implementation (CG-Exact). The experimental results show the benefits obtained for each of the presented methods and seem to indicate that, although not directly intended, the previously discussed strategies help to address the causes of slow convergence, *e.g.* the unstable behavior of dual variables values [10], diminishing the tail-off and heading-in effects.

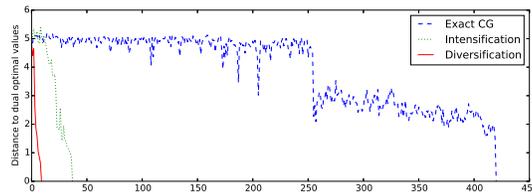


Fig. 1. Euclidean distance of dual variables to their optimal values along CG iterations

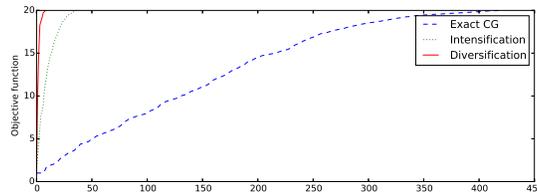


Fig. 2. Evolution of objective function along CG iterations

To help accelerate CG, it can be possible to combine CG with solution approaches that exploit several metaheuristics offering different performances both, in terms of the quality of the solution and in terms of the computational effort required for each. Furthermore, when these are used to return multiple and diverse columns, the number of iterations required to solve the problem might be heavily reduced while keeping CG being an exact method.

Bibliography

- [1] D. Adelman. Dynamic bid prices in revenue management. *Operations Research*, 55(4):647–661, 2007.
- [2] F. Castaño, A. Rossi, M. Sevaux, and N. Velasco. A column generation approach to extend lifetime in wireless sensor networks with coverage and connectivity constraints. *Computers & Operations Research*, 52:220–230, 2014.
- [3] M. Desrochers, J. Desrosiers, and M.M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, 40(2):342–354, 1992.
- [4] O. Du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194:229–237, 1999. ISSN 0012-365X.
- [5] Monica Gentili and Andrea Raiconi. α -coverage to extend network lifetime on wireless sensor networks. *Optimization Letters*, 7(1):157–172, 2013.
- [6] P.C. Gilmore and R.E. Gomory. A linear programming approach to the cutting-stock problem. *Operations research*, 9(6):849–859, 1961.
- [7] P.C. Gilmore and R.E. Gomory. A linear programming approach to the cutting stock problem-part ii. *Operations research*, 11(6):863–888, 1963.
- [8] E. Macambira, N. Maculan, and C. de Souza. Integer programming models for the sonet ring assignment problem. Technical report, Technical Report IC-05-026, Universidade Federal do Rio de Janeiro, 2005.
- [9] S.J. Maher. Solving the integrated airline recovery problem using column-and-row generation. *Transportation Science*, 2015.
- [10] R.E. Marsten, W.W. Hogan, and J.W. Blankenship. The boxstep method for large-scale optimization. *Operations Research*, 23(3):389–405, 1975.
- [11] N.T. Mounpla, L. Létocart, and A. Nagih. Solutions diversification in a column generation algorithm. *Algorithmic Operations Research*, 5(2):86–95, 2010.
- [12] P.J. Neame. Nonsmooth dual methods in integer programming phd thesis-department of mathematics and statistics. *The University of Melbourne March*, 1999.
- [13] A. Rossi, A. Singh, and M. Sevaux. Column generation algorithm for sensor coverage scheduling under bandwidth constraints. *Networks*, 60(3):141–154, 2012.
- [14] F. Vanderbeck. Implementing mixed integer column generation. In G. Desaulniers, J. Desrosiers, and M.M. Solomon, editors, *Column Generation*, pages 331–358. Springer US, 2005. ISBN 978-0-387-25486-9. 10.1007/0-387-25486-2_12.