The Lexicographic Minimum Gap Graph Partitioning Problem

Maurizio Bruglieri[1] and Roberto Cordone[2]

[1] Politecnico di Milano, Milan 20158, Italy,
maurizio.bruglieri@polimi.it,
[2] Universitá degli Studi di Milano, Milan 20133, Italy

Abstract. The *Minimum Gap Graph Partitioning Problem* consists in partitioning a vertex-weighted undirected graph into p connected subgraphs with minimum gap between the largest and the smallest vertex weight. In this work, we introduce the lexicographic version of the problem consisting in minimizing first the largest gap, then the sum of the gaps of all subgraphs. For this new variant we present a Ruin and Recreate solution approach, whose solution quality is assessed with the results provided by a mathematical programming formulation.

Keywords: graph partitioning, Ruin and Recreate metaheuristic, mathematical programming

1 Introduction

The *Minimum Gap Graph Partitioning Problem* (*MGGPP*) is a graph partitioning problem introduced in [3]. Let $G = (V, E)$ be an undirected graph, with $|V| = n$ vertices and $|E| = m$ edges, let $p \in N$ with $1 < p < n$ and let w be a *weight vector* which associates a weight $w_v \in N$ with each vertex $v \in V$. For each vertex subset $U \subseteq V$, we denote by $m_U = \min_{u \in U} w_u$ and $M_U = \max_{u \in U} w_u$ the minimum and maximum value of the weights in U, respectively. We denote by *gap* the difference $\gamma_U = M_U - m_U$, that corresponds to the maximum difference between the weights of the vertices in U. The *Minimum Gap Graph Partitioning Problem* (*MGGPP*) consists in finding a partition of G into p vertex-disjoint connected subgraphs $G_r = (V_r, E_r)$, with $r = 1, \ldots, p$, where all subgraphs must have at least two vertices (i.e., $|V_r| \geq 2$). Two versions of *MGGPP* have been introduced: the *min-sum* version minimizes the sum of all gaps, i.e., $\sum_{r=1}^{p} \gamma_{V_r}$ and the *min-max* version minimizes the largest gap, i.e., $\max_{r=1}^{p} \gamma_{V_r}$.

The *MGGPP* is motivated by applications in different fields. First, in the management of large Water Distribution Networks, it is a practice to sectorize them into subnetworks called District Metered Areas (*DMA*s) [4]. This allows to localize leakages more accurately, by monitoring the inflows and outflows of each district, and to better manage the pressure distribution, reducing the losses and the risk of damages. The optimal design of *DMA*s takes into account, among other objectives, the minimization of the difference between the required heads within the *DMA*s. The aim is to find a unique target pressure value in each *DMA*, and by consequence, to achieve an efficient pressure regulation also in networks with strong variations in ground elevation. If the network is represented as a graph where the edges correspond to pipes, the vertices to their intersections and the weight of a vertex to its ground elevation, the *MGGPP* models the search for an optimal sectorization. A second possible application concerns the levelling of farmlands, that makes irrigation and soil salt distribution more uniform, thus controlling weeds, saving water and increasing yield [6]. As it may be impractical or too expensive to flatten a sloping land as a whole, it is necessary to divide the land into parcels and build a flat terrace on each parcel by suitable earthworks. Choosing parcels with a small ground elevation difference helps to reduce the amount of ground to be moved, and therefore the cost associated with the corresponding earthworks. The *MGGPP* allows to model the division of a land into parcels with a limited difference in height (or also any other land attribute): the vertices of the graph correspond to sampled locations in the land, the weights to their heights, while the edges link adjacent locations.

Both the *min-sum* and the *min-max* version of *MGGPP* are in general NP-hardn and nonapproximable for every constant $\alpha < 2$, but tree graphs with a special structure (e.g., spiders and caterpillars) admit polynomial time algorithms [2]. In [1] a two-level Tabu Search algorithm and an Adaptive Large Neighborhood Search algorithm are proposed to solve the *MGGPP* in reasonable time. In this work we focus on a new variant of *MGGPP*, as explained in the following section.

2 The lexicographic *MGGPP*

In this work, we introduce the variant of the *MGGPP* where one wants to optimize both the objectives in a lexicographic way, i.e., first minimize the largest gap, then the sum of the gaps of all subgraphs. This is motivated by the drawbacks deriving from addressing one objective at a time. Indeed, the optimal solution with respect to the sum of all gaps very often consists in a subgraph with a large gap and several ones with small gaps; this solution is unbalanced. Vice versa, minimizing the largest gap, we usually obtain several optimal solutions with very different values for the gaps lower than the largest one. These solutions are not actually equivalent, since it would be desirable to minimize also their gaps.

3 A Ruin-and-Recreate solution method

Given the $\mathcal{NP}$-hardness of the problem, we propose a *ruin-and-recreate* (R&R) algorithm to solve real large instances. The R&R is an improvement heuristic originally proposed to solve traveling salesman, vehicle routing and network optimization problems [5] . Its main advantages are simplicity, effectiveness and suitability for complex optimization problems. The R&R consists in "ruining" a feasible solution, i.e., destroying a fraction of the solution and trying to restore feasibility as well as possible. The new feasible solution is later improved by a Local Search (LS) procedure.

In our application to the *MGGPP*, we ruin a feasible solution with four different removal procedures: i) the *random removal* eliminates from the current solution a fraction q of randomly selected vertices (respecting the minimum cardinality constraint on each subgraph); ii) the *worst cluster removal* removes all but two of the vertices of the cluster with the maximum gap; iii) the *Shaw removal* removes vertices that are topologically near or have similar weights (we expect it to be easier to create better solutions by reassigning similar vertices); iv) the *path removal* randomly selects a cluster and tries to get rid of the "worse" vertices by removing them together with paths going to the nearest cluster. In order to recreate a feasible solution, we apply four insertion heuristics that are more or less randomized versions of Prim's algorithm to build p connected components spanning the graph. The local search procedure adopted combines two different steepest descent mechanisms: i) moving a single vertex at a time from a subgraph to an adjacent one (provided that the original subgraph remains feasible); ii) merging together two subgraphs and breaking again the result into two subgraphs.

4 Some preliminary numerical results

The R&R algorithm has been implemented in C language. All experiments have been performed on an Intel Xeon E5-2620 2.1GHz server with 16GB of RAM and 16 cores. We considered 9 hydraulic networks provided by the Centre of Water Systems of the University of Exeter (https://emps.exeter.ac.uk/engineering/research/cws/resources/benchmarks/) from which we derive 27 instances of MGGPP considering 3 values of p. Comparing the results of R&R with those of a mathematical programming formulation solved by GUROBI we show that the former finds almost always the optimum, but in by far less CPU time.

References

1. Bruglieri, M., Cordone R.: Metaheuristics for the Minimum Gap Graph Partitioning Problem. *Computers & Operations Research*, vol. 132, 105301, (2021).
2. Bruglieri M., Cordone R., Lari I., Ricca F., Scozzari, A.: On finding connected balanced partitions of trees. *Discrete Applied Mathematics*, 299, pp. 1–16, (2021).
3. Bruglieri, M., Cordone R.: Partitioning a graph into minimum gap components. *Electronic Notes in Discrete Mathematics*, 55, pp. 33–36, (2016).
4. De Paola, F., Fontana, N., Galdiero, E., Giugni, M., Sorgenti degli Uberti, G., Vitaletti, M. Optimal design of district metered areas in water distribution networks, *Procedia Engineering*, 70, pp. 449–457, (2014)
5. Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., Dueck, G. Record breaking optimization results using the Ruin and Recreate principle, *Journal of Computational Physics* 159 (2), pp. 139–171, (2000)
6. Xiao, L., Hongpeng L., Dongling, N., Yan, W., Gang, L.: Optimization of GNSS-controlled land leveling system and related experiments. *Transactions of the Chinese Society of Agricultural Engineering*, 31 (3), pp. 48–55, (2015)