Adaptive Memory Programming for the Maximally Diverse Grouping Problem

Micael Gallego[1], Manuel Laguna[2], Rafael Martí[3], Abraham Duarte[1]

[1] Departamento de Ciencias de la Computación
Universidad Rey Juan Carlos, Spain. micael.gallego@urjc.es, abraham.duarte@urjc.es

[2] Leeds School of Business
University of Colorado at Boulder, USA. laguna@colorado.edu

[3] Departamento de Estadística e Investigación Operativa
Universidad de Valencia, Spain. rafael.marti@uv.es

Abstract

The maximally diverse grouping problem (MDGP) is an NP-hard problem that consists of forming groups from a given set of elements in such a way that the diversity in the groups is maximized. The MDGP has applications in academics, such as creating diverse teams of students, or in design of VLSI circuits. In this paper we propose a new procedure based on the tabu search methodology to obtain high quality solutions for this optimization problem. Specifically, our method includes a short-term memory component and makes use of the strategic oscillation. Our experimentation shows that the proposed methods compares favorably with previous metaheuristics.

1 Introduction

The maximally diverse grouping problem (MDGP) consists of grouping a given set of *M* elements into *G* mutually disjoint groups to maximize the diversity in each group. The diversity of each group is calculated as the sum of the distances between all the pairs of elements in the group, where the definition of the distances between elements is customized in each specific application. The objective of the MDGP is to maximize the sum of the diversity values of each group. This NP-hard problem [2] is also known as the *k*-partition problem [2] or the equitable partition problem [4]. The MDGP has a significant number of practical applications such as VLSI design of circuits [2], the academic context when forming diverse student groups [5] or creating diverse groups of peer reviews in scientific publications.

The MDGP has two variants depending on the size of the groups. In the first variant, MDGP1, the set of elements has to be divided into groups with the same size , where . In the second variant, MDGP2, the size of each group must fall in a prefixed interval where for . MDGP1 can be considered as a special case of MDGP2 for which for all . We target here the general case, MDGP2, simply referring to it as MDGP, and propose an adaptive memory programming method [3] to obtain high quality solutions.

2 Previous Methods

Weitz and Jelassi [5] developed WJ, a constructive method for solving MDGP, based on the idea of avoiding the assignment of similar elements to the same group. Weitz and Lakshminarayanan [6] adapted a previous improvement method and called it LC. It is based on exchanges. In each step, the method first selects an element lexicographically. Then, it identifies the group for which the diversity is maximized if is added to it. LC searches for the best exchange between and other element in and applies it only if the current solution improves. The method finishes when no further improvement is possible. LCW [6], is an evolution of LC in which the element to exchange with can belong to any group and is not limited to belong to the same group in which we are adding . Weitz and Lakshminarayanan [6] carried out extensive experimentation to compare all previous algorithms for the MDGP to conclude that random construction coupled with the LCW improvement method is the best procedure overall. Fan et al. [1] presented LSGA, a hybrid genetic algorithm coupled with the BI local search to solve the MDGP. BI follows the *best improvement* strategy, in which all possible interchanges are evaluated and the best one is finally applied (as opposed to the best strategy implemented in LCW). To the best of our knowledge, this is the first method for the general version of the MDGP (with different group sizes). Extensive experiments were conducted in [1] to compare LSGA with LCW starting from a random solution. These experiments showed the effectiveness of LSGA when solving MDGP instances with equal and different group sizes.

3 Tabu Search and Strategic Oscillation

In this section we describe the elements of our tabu search procedure for the MDGP. It basically consists of three elements: 1) construction of the initial solution, 2) neighborhood search and 3) strategic oscillation.

Our constructive method GC is based on a greedy evaluation. It starts by randomly selecting elements and assigning them to different groups. Then, GC performs iterations to assign the remaining unassigned elements to groups. The iterations are divided into two phases. Let be the set of elements currently assigned to group . In the first phase the unassigned elements are assigned to groups with and it finishes when all groups verify that . In the second phase, the unassigned elements are included in groups with and finishes when all elements are assigned to groups. In each iteration, consisting on the two phases described above, an unassigned element is randomly selected to be added to the group where the average distance between all its pairs of elements is maximized.

As described in Section 2, several improvement methods have been proposed for the MDGP where the LC, LCW and BI are the leading ones. We propose the improvement method FI, partially based on BI but implementing a *first improvement* strategy in the neighborhood exploration. FI iteratively evaluates the exchanges between elements and applies the first exchange that improves the objective value (instead the best one selected in BI).

The BI, FI, LC and LCW improvement methods are designed to solve the MDGP1. Starting from a feasible solution, they perform exchanges, always generating feasible solutions, until a local optima is reached. For the general problem, MDGP2, we extend the neighborhood exploration by adding *insertion* moves that allow transferring a single element from its current group to another group. We propose four new variants, T-BI, T-FI, T-LC and T-LCW that consider this extended neighborhood. We have also enhanced these improvement procedures by adding a short-term tabu memory in order to allow the search to continue beyond the first local optimum. Specifically, exchanged elements cannot be moved from their respective groups during *tabuTenure* iterations. The methods stop after performing *maxIter* consecutive iterations without improving the best solution found.

We have finally developed a *strategic oscillation* method [3], which relaxes the constraints of the problem for a few iterations with the objective of escaping from the local optimum. Strategic oscillation explores an enlarged search space including solutions where the group cardinality constraint may be violated. The oscillation between feasibility and infeasibility is defined by an integer parameter ranging between and . The method can explore infeasible solutions satisfying . To repair infeasible solutions we apply the method used in LSGA: randomly moving elements from groups with to groups with until all groups verify the cardinality constraints.

4 Computational Experiments

This section describes the computational experiments that we performed to test the effectiveness and efficiency of the procedures discussed above. We use 420 instances in our experimentation with different group configurations and sizes. Part of this set was previously introduced in [1] with . We have generated other types of instances including larger ones with .

In each experiment, we compute for each instance the overall best solution value, *BestValue*, obtained by the execution of all methods considered. Then, for each method, we compute the relative percentage deviation between the best solution value found by that method and the *BestValue*. We report the average of this relative percentage deviation (*Dev*) across all the instances considered in each particular experiment. We report, for each method, the number of instances (*#Best*) in which the value of the best solution obtained with this method matches *BestValue*. Finally, for multistart methods, we report (*#Const*) as the number of constructions.

In our first experiment we consider all types of instances with ranging from to totalizing 180 instances. We compare constructive methods coupled with local search for the MDGP. This experiment is configured using a factorial design where every constructive procedure (CG, WJ and a random construction) is coupled with every improvement method (BI, FI, LC and LCW). In order to test the contribution of the memory in the design of the algorithm, we also include in this experiment the tabu search variants (T-BI, T-FI, T-LC and T-LCW) with the following search parameters: and . Each multi-start procedure (i.e., either constructive + improvement or constructive + tabu) is executed for the same running time for each instance. The CPU time is limited according to the instance size: 1 second for instances with , 3 seconds for , 20 seconds for , 120 seconds for and 600 seconds for . This experiment concludes that GC coupled with T-LCW with and outperforms the rest of procedures. In the second experiment, we test the contribution of the strategic oscillation strategy to GC+T-LCW with different values of the parameter. This method is referred to as SO. Table 1 shows that there are no significant differences in quality between k_{max} values. We choose for SO since it presents the largest *#Best* statistic. In the third and final experiment with the 240 instances not used in the previous experiments, we compare our two best procedures (CG+T-LCW and SO) with LSGA [1], and LCW with random restarts [6]. Table 2 shows that SO clearly outperforms previous procedures in terms of quality (it exhibits a 0.04% average deviation while LCW and LSGA present 1.01% and 0.61% respectively). Moreover, the number of best known solutions found is 192 for SO while LCW and LSGA only obtain 80 and 82 respectively.

k_{max}	Dev	#Best	#Const
1	0.13%	111	1293.37
2	0.12%	100	758.43
3	0.13%	110	534.48
4	0.11%	112	415.19
5	0.10%	110	340.88

Table 1. parameter tuning in SO

Method	Dev	#Best	#Const
LCW	1.01%	80	-
LSGA	0.61%	82	7485.85
CG+T-LCW	0.17%	141	3152.72
SO	0.04%	192	338.55

Table 2. Best methods comparison

Conclusions

In this paper we present several heuristic procedures to solve the MDGP. The methods have been compared to state-of-the-art algorithms and the outcome of our experiments seems quite conclusive in regard to the merit of our procedure. We believe that the performance boost that we achieved by expanding search neighborhoods, by including additional moves, and search spaces and by allowing the search to visit infeasible solutions is a valuable lesson for future implementations.

References

[1] Z. P. Fan, Y. Chen, J. Ma and S. Zeng. A hybrid genetic algorithmic approach to the maximally diverse grouping problem. *Journal of the Operational Research Society*, 2010.

[2] T. A. Feo and M. Khellaf. A class of bounded approximation algorithms for graph partitioning. *Networks*, 1990, 20, 181-195.

[3] F. Glover, M. Laguna, Tabu Search. Kluwer Academic Publisher. 1997.

[4] F. A. O'Brien and J. Mingers. A heuristic algorithm for the equitable partitioning problem. *Omega*, 1997. 25(2): 215-223.

[5] R. R. Weitz and M. T. Jelassi. Assigning students to groups: a multi-criteria decision support system approach. *Decision Sciences*, 1992. 23(3): 746-757.

[6] R. R. Weitz and S. Lakshminarayanan. On a heuristic for the final exam scheduling problem. *The Journal of the Operational Research Society*, 1996. 47(4): 599-600.