An efficient Benders decomposition for the p-median problem

Cristian Duran-Mateluna[1,2,3], Zacharie Ales[1,2], Sourour Elloumi[1,2]

[1] UMA, ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France.
{cristian.duran, zacharie.ales, sourour.elloumi}@ensta-paris.fr

[2] CEDRIC, Conservatoire National des Arts et Métiers, Paris, France.

[3] PDSPS, Industrial Engineering Department, University of Santiago of Chile, Santiago, Chile.

Keywords : *discrete location, p-median problem, Benders decomposition.*

1 Introduction

The *p-median problem* (pMP) is one of the fundamental problems in Location Science [18]. In the (pMP), p sites have to be chosen from the set of candidate sites without considering set-up costs. The allocation costs are usually equal to the distance or travel time between clients and sites. The (pMP) is an NP-hard problem [17] and leads to applications where the sites may correspond to warehouses, plants, or shelters for example. This problem also occurs in the contexts of emergency logistics and humanitarian relief. A well-known clustering problem called *k-medoids problem* is also a special case of the (pMP) in which the set of clients and sites are identical. In this problem, sub-groups of objects, variables, persons, etc. are identified according to defined criteria of proximity or similarity.

A great interest in solving large location problems has led to the development of various heuristics and meta-heuristics in the literature. However, the exact resolution of large instances remains a challenge [19]. Some location problems have recently been efficiently solved using the Benders decomposition method (see e.g., [6, 10]). In this work, we explore a Benders decomposition of a recent formulation of the (pMP). We prove that the Benders cuts can be separated quickly. We implement a *branch-and-Benders-cut* approach that outperforms by an order of magnitude the state-of-the-art method of [11] on more than 200 benchmark instances with up to 238025 clients and sites. Our work is published in [8].

2 The p-median problem *(pMP)*

The p-median is formally defined as follows. Given a set of N clients, a set of M potential centers to open, and their corresponding index sets $\mathcal{N} = \{1, ..., N\}$ and $\mathcal{M} = \{1, ..., M\}$, let d_{ij} be the distance between client $i \in \mathcal{N}$ and site $j \in \mathcal{M}$ and let $p \in \mathbb{N}$ be the number of sites to open. The objective is to find a set S of p sites such that the sum of the distances between each client and its closest site in S is minimized. The (pMP) was introduced in [12] where the problem was defined on a graph such that a client can only be allocated to an open neighbor site. Since then, exact and approximation methods have been developed to solve the problem, as well as many of variants and extensions. We refer to [19] for more details and references.

There are three main MILP formulations for this problem. The classical one ($F1$) introduced in [24] that considers a binary variable y_j for each site $j \in \mathcal{M}$ that takes value 1 if the site is open, and 0 otherwise; and a binary variable x_{ij} that takes value 1 if client $i \in \mathcal{N}$ is allocated to site $j \in \mathcal{M}$ and 0 otherwise. An alternative formulation ($F2$) was proposed by [7] which for each client $i \in \mathcal{N}$ orders all its distinct distances to the sites. More formally, let K_i be the number of different distances from i to any site. Let $D_i^1 < D_i^2 < ... < D_i^{K_i}$ be these distances sorted, and let $\mathcal{K}_i$ be the corresponding index set $\{1, ..., K_i\}$. Formulation ($F2$) uses the same y variables as in ($F1$) and introduces new binary variables z. For any client $i \in \mathcal{N}$ and $k \in \mathcal{K}_i$,

z_i^k is equal to 0 if there is an open site at distance at most D_i^k from client i, and 1 otherwise. As $K_i \leq M$, it follows that $(F2)$ has at most as many variables and constraints as $(F1)$. Finally, [9] introduced formulation $(F3)$ based on $(F2)$. Given that, by definition, z_i^{k-1} is equal to 0 implies that z_i^k is also equal to 0, $(F3)$ replaces the following Constraints (1) of $(F2)$:

$$z_i^k + \sum_{j:d_{ij} \leq D_i^k} y_j \geq 1 \qquad i \in \mathcal{N},\ k \in K_i \tag{1}$$

by Constraints (4) and (5). Thus, the coefficients matrix of the following $(F3)$ is more sparse:

$$(F3) \begin{cases} \min & \sum_{i \in \mathcal{N}} \left(D_i^1 + \sum_{k=1}^{K_i - 1} (D_i^{k+1} - D_i^k) z_i^k \right) & & (2) \\ \text{s.t.} & \sum_{j \in \mathcal{M}} y_j = p & & (3) \\ & z_i^1 + \sum_{j:d_{ij} = D_i^1} y_j \geq 1 & i \in \mathcal{N} & (4) \\ & z_i^k + \sum_{j:d_{ij} = D_i^k} y_j \geq z_i^{k-1} & i \in \mathcal{N},\ k = 2, ..., K_i & (5) \\ & z_i^k \geq 0 & i \in \mathcal{N},\ k \in \mathcal{K}_i & (6) \\ & y_j \in \{0, 1\} & j \in \mathcal{M} & (7) \end{cases}$$

Since $(F3)$ is the most efficient formulation, we have studied a Benders decomposition based on this formulation. The main heuristics to solver the (pMP) are presented in the following surveys: [3, 14, 20, 21]. To the best of our knowledge, the state-of-the-art exact method is the `Zebra` algorithm [11], which considers a branch-and-cut-and-price algorithm based on formulation $(F2)$. It relies on the fact that the z variables satisfy $z_i^k \geq z_i^{k+1}$ in any optimal solution of $(F2)$ or its LP relaxation. Therefore, it is enough to solve these problems on a reduced subset of variables z, which is iteratively enlarged until an optimal solution is obtained. `Zebra` performs well on instances with up to 85900 nodes and large values of p.

3 Benders decomposition for the *(pMP)*

The Benders decomposition splits the optimization problem into a *master problem* and one or several *sub-problems*. The master problem and the sub-problems are solved iteratively and at each iteration, each sub-problem may add a cut to the master problem.

Based on $(F3)$, our master problem contains the location variables y and our sub-problem contains the allocation variables z. The master problem also contains a variable θ_i for each client $i \in \mathcal{N}$ which represents the allocation distances of the clients. Given a feasible solution of the master problem (a set of p sites), the sub-problem can be split into N sub-problems, each computing the allocation distance of one client and potentially leading to the addition of a cut. We show that Benders cuts can be quickly separated using the following definition.

Definition 1 *Given a solution $\bar{y}$ of the master problem from $(F3)$ or of its LP-relaxation, we define $\tilde{k}_i$ for each $i \in \mathcal{N}$ as follows:*

$$\tilde{k}_i = \begin{cases} 0 & \textit{if} \quad \sum_{j:d_{ij} = D_i^1} \bar{y}_j \geq 1 \\ \max\{k \in \mathcal{K}_i : \sum_{j:d_{ij} \leq D_i^k} \bar{y}_j < 1\} & \textit{otherwise} \end{cases}$$

Index $\tilde{k}_i$ is equal to 0 if client i is covered at distance D_i^1. Otherwise it is the largest distance index for which customer i is not yet covered in solution $\bar{y}$. Therefore, if $\bar{y}$ is binary, then the allocation distance of client i for this solution is $D_i^{\tilde{k}_i + 1}$. This definition allows us to state the following theorem.

Theorem 1 *Given a solution $\bar{y}$ of the master problem from $(F3)$ of its LP-relaxation and the corresponding indices $\tilde{k}_i$, the corresponding Benders cuts for each $i \in \mathcal{N}$ can be written as follows:*

$$\begin{cases} \theta_i \geq D_i^1 & \text{if } \tilde{k}_i = 0 \\ \\ \theta_i \geq D_i^{\tilde{k}_i+1} - \sum\limits_{j: d_{ij} \leq D_i^{\tilde{k}_i}} (D_i^{\tilde{k}_i+1} - d_{ij}) y_j & \text{otherwise} \end{cases}$$

This theorem ensures that there is a finite number of Benders cuts. This enables us to define the following new compact formulation $(F4)$ for the p-median problem.

$$(F4) \begin{cases} \min & \sum\limits_{i \in \mathcal{N}} \theta_i \\ \text{s.t.} & \sum\limits_{j \in \mathcal{M}} y_j = p \\ & \theta_i \geq D_i^1 & i \in \mathcal{N} & (8) \\ & \theta_i \geq D_i^{k+1} - \sum\limits_{j: d_{ij} \leq D_i^k} (D_i^{k+1} - d_{ij}) y_j & i \in \mathcal{N},\ k \in \{1, ..., K_i - 1\} & (9) \\ & y_j \in \{0, 1\} & j \in \mathcal{M} \end{cases}$$

Constraints (8) correspond to the lower bound of each allocation distance given by the smallest distance value from each client to its nearest site. Constraints (9) ensure that each variable θ_i is larger than D_i^{k+1} unless a site j is opened at a distance not more than D_i^k from client i. In this case the allocation distance of client i is at most d_{ij}.

The efficiency of our decomposition comes from an $O(M)$ algorithm to separate the Benders cuts along with several implementation improvements in a two-phase resolution. In Phase 1, the integrity constraints are relaxed, which allows many cuts to be generated quickly. In order to enhance the performance of this phase, we provide a good candidate solution to the initial master problem which significantly reduces the number of iterations. To that end, we use the state-of-art `PopStar` heuristic [23]. Additionally, we use a rounding heuristic at each iteration to try to improve the upper bound of the problem. At the end of Phase 1, most of the generated Benders cuts are not saturated by the current fractional solution. We remove most of them to reduce the number of constraints in the master problem. Phase 1 provides a lower and an upper bound of the problem, which can be used to perform an analysis of the reduced costs of the last fractional solution to reduce the number of variables. In Phase 2, we add the integrity constraints and the obtained master problem is solved through a *branch-and-Benders-cut*. At each node which provides an integer solution, we solve the sub-problems in order to generate Benders cuts. The resolution of the sub-problems is performed through callbacks which is a feature provided by mixed integer programming solvers.

4 Computational study

Our study was carried out on an Intel XEON W-2145 processor 3,7 GHz, with 16 threads, but only 1 was used, and 256 GB of RAM. IBM ILOG CPLEX 20.1 was used as the *branch-and-cut* framework. The absolute tolerance to the best integer objective (EpGap) has been set to 10^{-10}, and the absolute MIP gap (EpAGap) to 0.9999.

We study the same instances used for the state-of-the-art method `Zebra` that is the benchmark instances from OR-Library [4] and TSP-Library [22]. In all these instances, the sites are at the same location as the clients and thus $N = M$. The set of OR-Library contains instances with 100 to 900 clients, and the value of p is between 5 and 500. We select a set from the TSP-Library having between 1304 and 238025 clients. Another set of symmetric instances that satisfy triangle inequality are the BIRCH instances, usually solved by heuristics algorithms (see e.g [1, 13, 16]). We consider two types of instances with sizes from 10000 to 20000 points for the first one and from 25000 to 89600 points for the second one. We compare performances of our algorithm to that of an aggregation heuristic [1] (denoted by `AvellaHeu`) for the first type, and an hybrid heuristic combining aggregation and variable neighborhood search [15] (denoted by `IrawanHeu`) for the second type. We also consider the RW instances originally proposed by [23] with the `PopStar` heuristic. They correspond to completely random distance matrices. The distance between client i and site j is not necessarily equal to the distance between site j and client i. Four different values of $N = M$ are considered: 100, 250, 500, and 1000. We also consider the ODM instances which were introduced by [5] and are solved in [2] with a *branch-and-cut-and-price* algorithm (denoted by `AvellaB&C`). These instances correspond to the *optimal diversity management problem* in which certain allocations between clients and sites are not allowed. We consider the hardest instances in which $N = 3773$.

To summarize all our results, Table 4 presents for each dataset the number of instances considered, the average cpu time ($\overline{Time}$) computed on the instances solved to optimality by both approaches, the percentage of optimally solved instances (*Opt*), and the average final optimality gap ($\overline{Gap}$). The grey values indicate that the solution times were not obtained on the same computer, unlike `Zebra` and `PopStar`. We can see that our solution times are significantly faster than the two other exact approaches and are competitive with the ones of the heuristics. Moreover, the number of instances that we solve to optimality and our average gaps are significantly better.

Dataset	# instances	Algorithm	$\overline{Time}$	*Opt*	$\overline{Gap}$
TSP	149	`Zebra`	2408s	68%	0.28%
		Our method	**381s**	**91%**	**0.03%**
BIRCH dsx	24	`AvellaHeu`*	159s	79%	0.01%
		Our method	**95s**	**100%**	**0%**
BIRCH dsn	24	`IrawanHeu`*	1965s	8%	0.32%
		Our method	**766s**	**100%**	**0%**
RW	27	`PopStar`*	**6s**	48%	9.47%
		Our method	15s	**59%**	**6.90%**
ODM	10	`AvellaB&C`	147248s	30%	1.94%
		Our method	**560s**	**100%**	**0%**

TAB. 1: Summary of all our results. * means heuristic method

We highlight that several instances are solved for the first time having up to 89600 and 238025 clients and sites from the BIRCH and TSP libraries, respectively. For the RW instances, it was possible for the first time to solve instances with up to 1000 clients with a large value of p. For ODM instances with 3773 clients, previously unsolved instances were solved within 10 hours. To give more detailed results, Table 2 presents the results obtained on the largest TSP instances. In which column OPT/BKN contains the optimal value of the instance in **bold** if it is known or the best-known solution value obtained given the time limit, otherwise. If the value is <u>underlined</u>, it means that it is the first time the instance is solved to optimality or that the best-known value was improved. We observe that we are significantly faster than `Zebra`. Moreover, when both methods reach the time limit, we are able to prove significantly smaller gaps.

5 Conclusions

We define a Benders decomposition based on the most efficient formulation of the p-median problem. The efficiency of the proposed decomposition comes from a fast algorithm for the sub-problems in conjunction with additional improvements implemented in a two-phase algorithm. In the first phase, the integrity constraints are relaxed and in the second phase, the problem is solved in an efficient *branch-and-cut* approach.

Our approach outperforms other state-of-the-art methods on five data sets from the literature. Our approach was able to improve the best-know solution of 91% of the instances which had not previously been solved to optimality. Finally we found an optimal solution for 81% of them. One of the perspectives of this research is to exploit these results on other families of location problems. It is also expected to use other branching strategies that could allow a greater efficiency during the development of the *branch-and-cut* algorithm.

INSTANCE				PHASE 1			PHASE 1 + 2				Zebra	
name	$N = M$	p	*OPT/ BKN*	LB^1	UB^1	$t^1[s]$	*gap*	*iter*	*nodes*	$t^{tot}[s]$	*gap*	$t[s]$
ch71009	71009	10000	4274662	4273680	4424131	6326	0,006%	36	18585	TL	∞	♦
		20000	**2377760**	2377409	2419539	681	0%	40	474	3581	∞	♦
		30000	**1464151**	1464015	1473517	431	0%	27	0	819	∞	♦
		40000	**879336**	879272	881997	220	0%	17	0	465	∞	♦
		50000	**463553**	463544	463904	133	0%	24	0	258	0%	653
		60000	**167565**	167558	167789	49	0%	31	0	135	0%	331
pla85900	85900	10000	166853134	166627292	182428500	2841	0,12%	30	2113	TL	∞	♦
		20000	109007210	107246411	120645337	3975	1,58%	27	618	TL	∞	♦
		30000	86944862	86944715	87547287	1411	0,0002%	84	28033	TL	∞	♦
		40000	**69944715**	69944715	69965668	1006	0%	12	0	1006	∞	♦
		50000	**52944715**	52944715	52945623	921	0%	12	0	921	∞	♦
		60000	**35944715**	35944715	35945105	858	0%	11	0	858	∞	♦
		70000	**18977475**	18977475	18977475	73	0%	13	0	73	0%	122
		80000	**4512752**	4512752	4512752	12	0%	20	0	13	0%	97
usa115475	115474	20000	5287343	5286659	5383798	3366	0,001%	36	11102	TL	∞	♦
		30000	**3815620**	3815143	3861590	1494	0%	41	589	11581	∞	♦
		40000	**2876909**	2876603	2904492	1353	0%	32	459	4431	∞	♦
		50000	**2189144**	2188903	2200969	1122	0%	28	480	3189	∞	♦
		60000	**1651400**	1651234	1657118	795	0%	25	0	1588	∞	♦
		70000	**1214299**	1214177	1217251	612	0%	17	0	1045	∞	♦
		80000	**851481**	851422	852851	435	0%	24	0	788	∞	♦
		90000	**548097**	548076	548560	270	0%	18	0	544	∞	♦
ara238025	238025	10000	1354335	1345698	1446100	5197	0,64%	19	0	TL	∞	♦
		20000	857553	857453	878372	5582	0,004%	42	696	TL	∞	♦
		30000	**630969**	630872	643171	5123	0%	33	663	33687	∞	♦
		40000	**494842**	494804	498378	4135	0%	18	0	10028	∞	♦
		50000	**401835**	401795	404218	2675	0%	19	0	8327	∞	♦
		60000	**334279**	334236	335807	2969	0%	17	0	7240	∞	♦
		70000	**283627**	283592	286065	3058	0%	28	509	19298	∞	♦
		80000	**244233**	243936	248742	2578	0%	36	378	14615	∞	♦
		90000	**214233**	213936	219673	1548	0%	27	507	28391	∞	♦
		100000	**184233**	184069	188556	1973	0%	44	613	18473	∞	♦
		150000	**88025**	88025	88334	1532	0%	27	0	6057	∞	♦
		200000	**38025**	38025	38025	319	0%	11	0	319	∞	♦

TAB. 2: Results on the largest TSP instances for our method and `Zebra` on our computer. TL=36000 seconds. ♦ means that the computer ran out of memory.

References

[1] P. Avella, M. Boccia, S. Salerno, and I. Vasilyev. An aggregation heuristic for large scale p-median problem. *Computers & Operations Research*, 39(7):1625–1632, 2012.

[2] P. Avella, A. Sassano, and I. Vasil'ev. Computational study of large-scale p-median problems. *Mathematical Programming*, 109(1):89–114, 2007.

[3] S. Basu, M. Sharma, and P. S. Ghosh. Metaheuristic applications on discrete facility location problems: a survey. *OPSEARCH*, 52(3):530–561, 2015.

[4] J. E. Beasley. Or-library: Distributing test problems by electronic mail. *The Journal of the Operational Research Society*, 41(11):1069–1072, 1990.

[5] O. Briant and D. Naddef. The optimal diversity management problem. *Operations research*, 52(4):515–526, 2004.

[6] J.-F. Cordeau, F. Furini, and I. Ljubić. Benders decomposition for very large scale partial set covering and maximal covering location problems. *European Journal of Operational Research*, 275(3):882 – 896, 2019.

[7] G. Cornuejols, G. L. Nemhauser, and L. A. Wolsey. A canonical representation of simple plant location problems and its applications. *SIAM Journal on Algebraic Discrete Methods*, 1(3):261–272, 1980.

[8] C. Duran-Mateluna, Z. Ales, and S. Elloumi. An efficient benders decomposition for the p-median problem. *European Journal of Operational Research*, 308(1):84–96, 2023.

[9] S. Elloumi. A tighter formulation of the p-median problem. *Journal of Combinatorial Optimization*, 19(1):69–83, 2010.

[10] M. Fischetti, I. Ljubic, and M. Sinnl. Redesigning benders decomposition for large-scale facility location. *Management Science*, 63(7):2146–2162, 2017.

[11] S. García, M. Labbé, and A. Marín. Solving large p-median problems with a radius formulation. *INFORMS Journal on Computing*, 23(4):546–556, 2011.

[12] S. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12:450–459, 1964.

[13] P. Hansen, J. Brimberg, D. Urošević, and N. Mladenović. Solving large p-median clustering problems by primal–dual variable neighborhood search. *Data Mining and Knowledge Discovery*, 19(3):351–375, 2009.

[14] C. Irawan and S. Salhi. Aggregation and non aggregation techniques for large facility location problems: A survey. *Yugoslav Journal of Operations Research*, 25:1–1, 01 2015.

[15] C. A. Irawan and S. Salhi. Solving large p-median problems by a multistage hybrid approach using demand points aggregation and variable neighbourhood search. *Journal of Global Optimization*, 63(3):537–554, 2015.

[16] C. A. Irawan, S. Salhi, and M. P. Scaparra. An adaptive multiphase approach for large unconditional and conditional p-median problems. *European Journal of Operational Research*, 237(2):590–605, 2014.

[17] O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979.

[18] G. Laporte, S. Nickel, and F. Saldanha-da Gama. *Location Science (2nd ed)*. Springer International Publishing, 2019.

[19] Marín and M. Pelegrín. *The p-Median Problem*, pages 25–50. Springer International Publishing, 01 2019.

[20] N. Mladenović, J. Brimberg, P. Hansen, and J. A. Moreno-Pérez. The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 2007.

[21] J. Reese. Solution methods for the p-median problem: An annotated bibliography. *Networks*, 48(3):125–142, 2006.

[22] G. Reinelt. TSPLIB—A Traveling Salesman Problem Library. *ORSA Journal on Computing*, 3(4):376–384, 1991.

[23] M. G. C. Resende and R. F. Werneck. A Hybrid Heuristic for the p-Median Problem. *Journal of Heuristics*, 10(1):59–88, 2004.

[24] C. S. ReVelle and R. W. Swain. Central facilities location. *Geographical Analysis*, 2(1):30–42, 1970.