Solving a hierarchical network design problem with two stabilized column generation approaches

Mathieu Trampont* Christian Destré* Alain Faye◇

*Orange Labs
38-40 rue du Général Leclerc 92740 Issy-les-Moulineaux, France

◇CEDRIC Laboratory, Conservatoire National des Arts et Métiers, France
292 Rue St Martin 75141 Paris Cedex 03, France

Abstract

In this paper, we focus on a variant of the Multisource Weber problem. In this classical continuous location problem, the location of a fixed number of concentrators, and the allocation of terminals to them, must be chosen to minimize the total cost of links between terminals and concentrators. In our variant, we have a third hierarchical level, two categories of link costs, and the number of concentrators to locate is unknown. This difficult problem arises in telecommunications, and we propose some column generation approaches to deal with large scale instances.

Keywords: *Multisource Weber Problem, Central Cutting Plane, Column Generation*

A 3-level hierarchical network design problem

In this paper, we focus on a variant of the multisource Weber problem [1, 3]. In our problem, we have a set of terminals we want to connect to a central equipment through some local concentrators. Each terminal is to be connected to one concentrator and all the concentrators are to be connected to the central equipment. We have to choose the number of concentrators and their locations, while minimizing the costs of the links between the terminals, the concentrators, and the central equipment. Concentrators are uncapacitated. Without central equipment and with a known number of concentrators, this problem is called a multisource Weber problem . Since we do not know the number of concentrators, we consider their installation costs (same constant C for each concentrator). Another specificity of our problem is the two different costs for each link:

- **A fixed cost** related to the link infrastructure. It is proportional to the link length.
- **A variable cost** related to the cables connecting terminals to the central equipment. As the fixed cost, it is proportional to the length of the link, but also to the number of demands passing through the link.

A_1 and B_1 (respectively A_2 and B_2) denote the variable and fixed costs per meter between terminals and their concentrator (respectively concentrators and the central equipment). V_{y_i} denote the set of terminals linked to concentrator y_i, and $\|x - y\|$ the Euclidean distance between two points x and y. The network architecture and the associated costs are described in Figure 1 for terminals with unit demands. In real networks, B_2 is far more important than A_2, so it is only interesting to solve this problem for large dense instances with many terminals per concentrators.

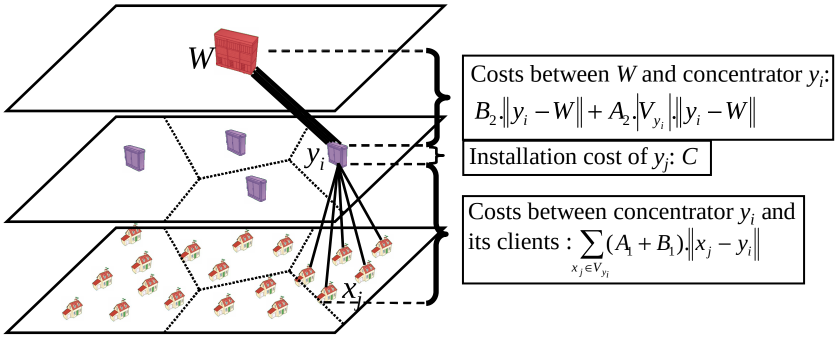

Figure 1: The three hierarchical levels with their costs

This variant remains close to the multisource Weber problem. Methods used to solve the latter can be used for the former. We had previously developed heuristics adapted from [1] or using stochastic geometry [8, 7], but we missed a lower bound to assess the quality of our solutions. To obtain this bound, we turned toward column generation approaches for our problem. In this paper, we propose a new stabilized column generation approach, based on a central cutting planes method [4, 5]. We compare the results of this new approach to a classical column generation method similar to the one used in [3] to solve the multisource Weber problem.

The modelling for classical column generation

In the usual modelling for the multisource weber problem, the coordinates of the concentrators are part of the variables, and the Euclidean distance between concentrators and terminals appears in the objective function. This is not very convenient since this objective function is neither convex nor concave. Instead, our problem can be modelled as a set partitioning problem. Each variable represents a group of terminals. The problem is then to decide which group of terminals should be assigned the same concentrator. The cost of each group sums the installation cost of its concentrator, the costs of links between its concentrator and its terminals, and the cost of the connection with the central equipment. To obtain the cost c_i of a given group i, we have to optimally locate its concentrator by solving the following Weber problem:

$$c_i = C + \min_{(\mathbf{r}_i,\mathbf{s}_i)} \Big[\sum_{j=1}^{n} b_{ij} \Big((w_j A_1 + B_1) d(i,j) + w_j A_2 D(i,W) \Big) + B_2 D(i,W) \Big] \tag{1}$$

where:

- $(\mathbf{r}_i, \mathbf{s}_i)$ are the coordinates of the concentrator assigned to group i;
- b_{ij} is a binary coefficient denoting terminal j membership to group i (*i.e.* b_{ij} equals 1 if j is in group i, 0 otherwise);
- w_j is the weight representing the demand of terminal j.
- $d(i,j)$ is the Euclidean distance between terminals j and the concentrator assigned to group i;
- $D(i,W)$ is the Euclidean distance between the central equipment W and the concentrator assigned to group i;

Using these definitions we obtain the following modelling:

$$(\mathcal{P}) \begin{cases} \min\limits_{\mathbf{z}} \quad \sum\limits_{i=1}^{|\mathcal{P}(E)|-1} \Big(c_i \mathbf{z}_i \Big) \\ \text{s.t.} \quad \sum\limits_{i=1}^{|\mathcal{P}(E)|-1} b_{ij}\mathbf{z}_i \geq 1 \quad \forall j = 1 \dots n \\ \mathbf{z}_i \in \{0,1\} \end{cases} \tag{2}$$

where E is the terminals set of size n and $\mathcal{P}(E)$ the set containing all subsets of E. The elements of $\mathcal{P}(E)$ are indexed by i. $\mathbf{z}_i$ is a decision variable which equals 1 when terminals group i is used, 0 otherwise. Constraints (2) ensure that each terminal is covered by at least one chosen group. Compared to the one in [3], this formulation has different costs c_i and no constraint limiting the number of chosen groups.

In the following, we focus only on solving the continuous relaxation $(\mathcal{P}')$ of $(\mathcal{P})$. The principle of column generation is to solve a restricted problem $(\mathcal{RP})$ with a subset I of all the variables $\mathbf{z}_i$, since only a few of these variables are used in the optimal solution. Then we use the optimal dual values $\mathbf{u}_j$ of the restricted problem to find variables with a negative reduced cost. These useful variables are added to $(\mathcal{RP})$ which is solved again, until no more useful variables can be found. In this way, we obtain the optimal solution of $(\mathcal{P}')$.

The problem of finding the group with the most negative reduced cost (called the subproblem) is stated as follows:

$$(\mathcal{SP})\begin{cases} \min\ \bar{c}(\mathbf{b},\mathbf{X}) = C + \Big[\sum_{j=1}^{n}\mathbf{b}_j\Big((w_jA_1+B_1)d(\mathbf{X},j) + w_jA_2D(\mathbf{X},W) - u_j\Big) \\ \qquad\qquad\qquad + B_2D(\mathbf{X},W)\Big] \end{cases}$$

where: the variable $\mathbf{X}$ denotes the location point of the concentrator, $\mathbf{b}_j$ is a decision variable which equals 1 if terminal j is added to the group, and u_j is the dual value associated with the j-th constraint of type (2). In the multisource Weber problem, the terms corresponding to the cost of the link between the concentrator and the central equipment do not appear. In that case, the subproblem can be solved as a Weber problem with limited distances [2, 3]. Our subproblem is different, yet we use an algorithm similar to the one used in [2] to solve it. First, we point out that there exists an optimal solution $\left(b_j^*, X^*\right)$ which satisfies the following property:

$$\forall j = 1\dots n, \qquad b_j^* = 1 \Leftrightarrow (w_jA_1+B_1)d(X^*,j) + w_jA_2D(X^*,W) - u_j \le 0 \tag{3}$$

The curve defined by the following equation:

$$(w_jA_1+B_1)d(X^*,j) + w_jA_2D(X^*,W) = u_j \tag{4}$$

is called a Cartesian oval. It has two focuses: terminal j and the central equipment W. In particular, when $w_jA_1 + B_1 = w_jA_2$, the curve is an ellipse. So the property (3) means that terminal j is used in the optimal solution if and only if the concentrator lies into the Cartesian oval with terminal j and W as focuses.

In the Weber problem with limited distances, there is a property similar to (3) with circles centred on each terminal, instead of Cartesian ovals. Cartesian ovals can be thought as circles centred on a terminal and “stretched” towards the central equipment. The algorithm proposed in [2] is based on the idea that the number of potentially optimal solutions verifying the property is bounded by a polynomial of the number of terminals. Each of these solutions corresponds to a combination of terminals, which itself corresponds to a region of the plane split by the circles. Theses regions can be listed by means of the intersection points between the circles. Similar ideas can be used with Cartesian ovals. Computing the intersection points of two Cartesian ovals is slightly more difficult but doable. Using the same arguments used in [2], we can conclude that the number of potentially optimal solutions is bounded by $8n(n-1)$. To find the optimum of $(\mathcal{SP})$, each potential solution is evaluated by solving a simple Weber problem.

Being able to solve the subproblem to optimality, we implemented a classical column generation algorithm to solve the continuous relaxation of our problem. We used solutions of heuristics to provide initial columns. Since we faced the same degenerative behaviours as in [3], we implemented the same stabilization techniques, *i.e.* boxstep method and interior point stabilization method of [6]. In addition to this classical column generation, we developed a new one based on a central cutting plane algorithm. This new approach is described in the next section.

A geometric centre column generation approach

Cutting planes methods are the dual counterpart of column generation. Adding the column with the most negative reduced cost in the primal corresponds to adding the most violated cut in the dual. Central cutting planes algorithms tend to be more stabilized than the one corresponding to classical column generation. That's why we used the cutting planes algorithm presented in [5], which is an acceleration of the central cutting planes algorithm of Elzinga and Moore [4], to solve our problem. This algorithm solves a different restricted problem. To explain it we must consider the dual problem $(\mathcal{DRP})$ of $(\mathcal{RP})$:

$$(\mathcal{DRP})\begin{cases}\max\limits_{\mathbf{u}} \quad \sum_{j=1}^{n}\mathbf{u}_j \\ \text{s.t.} \sum_{j=1}^{n} b_{ij}\mathbf{u}_j \le c_i \quad \forall i = 1\ldots|\mathcal{I}| \quad (5) \\ \mathbf{u}_j \ge 0\end{cases}$$

where I is the reduced set of terminals groups. The principle of the Elzinga and Moore's algorithm is to find the centre of the largest sphere contained in a polyhedron consisting of the inequalities (5) and a lower bound inequality. The polyhedron must be bounded for the shere radius to be finite. For our problem, this is easily achieved by initializing the problem with basics columns containing only one terminal. There are different ways to use the lower bound: [4, 5] propose two different inequalities, we propose a third. The restricted problem becomes $(\mathcal{RQ})$:

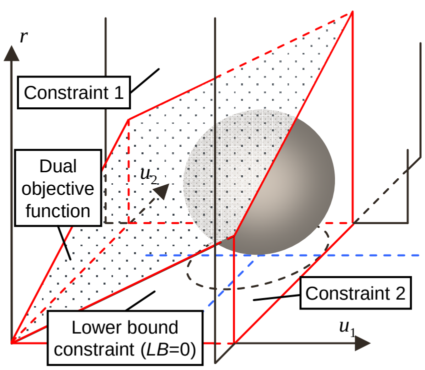

Figure 2: Geometrical interpretation of the linear program $(\mathcal{RQ})$.

$$(\mathcal{RQ})\begin{cases}\max\limits_{\sigma,\mathbf{u}} \quad \sigma \\ \text{s.t.} \\ \quad \sum_{j=1}^{n} b_{ij}\mathbf{u}_j + \sigma\sqrt{\sum_{j=1}^{n} b_{ij}^2} \le c_i \qquad \forall i = 1\ldots|\mathcal{I}| \quad (6) \\ \sum_{j=1}^{n}\mathbf{u}_j - \sigma(1+\sqrt{n+1}) \ge LB \qquad (7) \\ \mathbf{u}_j \ge 0\end{cases}$$

where σ is the radius of the sphere, $\mathbf{u}$ is the vector of coordinates of the centre of the sphere in the dual space, and LB is the lower bound of our problem. Inequalities (6) ensure that any point of the sphere respects constraints (5), as illustrated with constraints 1 and 2 in figure 2. Inequality (7) ensures that any point of the sphere respects the following inequalities: $\sum_{j=1}^{n}\mathbf{u}_j \ge \mathbf{r}$ and $\mathbf{r} \ge LB$ which mean that every point is between the lower bound and the objective function of $(\mathcal{DRP})$, as in figure 2.

Once this restricted problem is solved, the optimal sphere centre coordinates are used in the subproblem. If $(\mathcal{SP})$ returns no violated inequalities, the centre is a dual admissible solution and provides an improvement of the lower bound LB. The Elzinga and Moore's algorithm continue to solve $(\mathcal{RQ})$ and $(\mathcal{SP})$ until the radius σ is smaller than a chosen ϵ.

Betrò proposed in [5] to accelerate this algorithm by making assumptions on the lower bound. If you have an upper bound UB and a lower bound LB for your problem, Betrò replaced the inequality (7) by

$$\sum_{j=1}^{n}\mathbf{u}_j - \sigma(1+\sqrt{n+1}) \ge \tau \qquad (8)$$

where τ is a convex combination of LB and UB. τ may not be a valid lower bound. In this case, at some point during the algorithm, $(\mathcal{RQ})$ becomes infeasible. We know then that τ is in fact an upper bound. UB and τ can be updated. For example, when $\tau = \frac{UB+LB}{2}$, Betrò's algorithm consists of splitting in two the gap, each time $(\mathcal{RQ})$ is infeasible or $(\mathcal{SP})$ returns no column. The algorithm stops when the difference between UB and LB is smaller than a chosen ϵ.

Using Betrò's algorithm does not prevent heavy degenerative behaviour. If $(\mathcal{RP})$ can present an infinity of optimal dual solutions, $(\mathcal{RQ})$ can as well present an infinity of optimal spheres. That's why we tried to adapt stabilization techniques that we used on $(\mathcal{RP})$ for $(\mathcal{RQ})$. We first considered the interior point method of [6]. It is very easy to implement with $(\mathcal{RQ})$. After solving $(\mathcal{RQ})$ once with an optimal radius value σ^*, σ is fixed to σ^* and the objective is changed to

$$\max_{\mathbf{u}} \quad \sum_{j=1}^{n} \omega_j \mathbf{u}_j \quad \text{and/or} \quad \min_{\mathbf{u}} \quad \sum_{j=1}^{n} \omega_j \mathbf{u}_j$$

where the coefficients ω_j are randomly chosen between 0 and 1. We chose to solve 19 modified problems, so we obtain at most 20 different optimal centres. Their mean is used in $(\mathcal{SP})$.

For boxstep method, it is more complex. The principle of boxstep method is to limit with bounds the movement in the dual space, from one iteration to another. The box centre is called stability centre and is usually the current best dual admissible solution. Its coordinates in the dual space are denoted by v_j and the size of the box by d. This leads us to add the following constraints in $(\mathcal{RQ})$.

$$\mathbf{u}_j \leq v_j + d \qquad\qquad \mathbf{u}_j \geq v_j - d \tag{9}$$

These constraints complicate the course of the algorithm. When $(\mathcal{RQ})$ becomes infeasible, we have to determine if it is caused by the box or by τ. To do this, we first remove the box constraints (9) and see if the problem is still infeasible. In this case, UB and τ are updated, otherwise we put back the box constraints and increase the box size until the problem becomes feasible. Another issue appears when no cuts are violated. The stability centre is then updated to the sphere centre and τ is increased, which can easily induce infeasibility if the box size is lower than the increase. To prevent this, we increase the box size by the same amount as τ, each time the stability centre is updated.

After presenting the algorithm principles, we compare the performances of the two column generation methods, and also of the former multiphase heuristic.

Computational results and conclusion

To compare the two column generation methods, the same parameters were chosen for interior point and boxstep stabilization techniques. 20 restricted problems are solved for the purpose of interior point technique. For boxstep technique, we increase the box size by multiplying it by 1.5 (except when τ is increased, as explained before). Every time the restricted problem is feasible and the subproblem returns some columns to add, box size is multiplied by 0.9. Initializations of the stability centre and box size are the same for both methods.

Size	515	508	530	492	561	468	536	504	562	485	516.1
Classical col. gen. (s)	2951	1548	1196	2516	2971	1142	2118	2045	2059	1963	2050.9
Central col. gen. (s)	555	627	495	691	853	575	568	433	751	519	606.7
Heuristic gap	0.12%	0.07%	0.08%	0.09%	0.09%	0.06%	0.01%	0.004%	0.07%	0.04%	0.06%
Weaker initialization											
Classical col. gen. (s)	2403	1362	2718	2117	4069	1108	2306	2280	2650	1168	2218.1
Central col. gen. (s)	1022	623	673	455	3003	367	717	573	729	587	874.9

Table 1: CPU time used on ten instances

In table 1, we present computational results obtained on one set of ten instances with an average size of more than 500 terminals. These instances were obtained with stochastic geometry (see [7, 8]). In the first part of the table, the column generation was initialized with many columns from the heuristics solutions (the solution of the multiphase heuristic and 50 local optima from stochastic geometry). For each instance are reported the CPU time used and the gap with the multiphase heuristic solution. In the second part of table 1, we compared CPU time for the second set of instances with a lighter initialization (only 10 local optima from stochastic geometry). It shows that the lack of initial information has only a small impact on the two column generation methods.

Finally, we used the central column generation on a real-like instance with more than $1,700$ terminals. The CPU time used was too long and we had to stop the algorithm. When we stopped the resolution, the current lower bound showed that the multi-phase heuristic was within 10% of the optimal value.

These results shows that the multiphase heuristic adapted from [1] is an excellent way to achieve a trade-off between solution quality and CPU time consumption, since its solutions have an average gap of less than 0.1% and column generation methods are still much slower (the multiphase heuristic takes less than a minute in average to solve these instances).

To come to these results, we implemented column generation algorithms for our problem, solving a new subproblem on the occasion. We can see that our implementation of a column generation method based on a central cutting plane algorithm with usual stabilization techniques, outperforms the classical one, with a mean CPU time more than 3 times lower. This method proved itself to be appropriate when dealing with the degenerative behaviors of our problem, and we believe further studies will show its usefulness on other ones.

References

[1] J. Brimberg, N. Mladenovic, S. Salhi, *The multi-source Weber problem with constant opening cost*, Journal of the Operational Research Society, 55, 640-646, 2004.

[2] Z. Drezner, A. Mehrez, G. O. Wesolowsky, *The Facility Location Problem with Limited Distances*, Transportation Science, Vol.25, No.3, 183-187, 1991.

[3] G. Righini et L. Zaniboni, *A branch-&-price algorithm for the multi-source Weber problem*, Int. J. Operational Research, Vol.2, No.2, 188-207, 2007.

[4] J. Elzinga et T. G. Moore, *A central cutting plane algorithm for the convex programming problem*, Mathematical Programming, Vol.8, 134-145, 1975.

[5] B. Betrò, *An accelerated central cutting plane algorithm for linear semi-infinite programming*, Mathematical Programming, Vol.101, 479-495, 2004.

[6] L-M. Rousseau, M. Gendreau, D. Feillet, *Interior point stabilization for column generation*, Operations Research Letters, Vol. 35, 660-668, 2007

[7] F. Baccelli and S. Zuyev, *Poisson-Voronoi spanning trees with applications to the optimization of communication networks*, Operations Research, 47(4), 619-631, 1999.

[8] M. Trampont and C. Destré, *Two approaches for solving a continuous location problem: Stochastic geometry and Operational Research*, INOC 2007.