Advances in Decision Technology and Intelligent Information Systems, Volume IV, K.J. Engemann and G.E. Lasker, Eds.
The International Institute for Advanced Studies in Systems Research and Cybernetics, 2003, pp. 16-20

Algorithms for Optimal Switch Location: Concave Cost Functions

Boris S. Verkhovsky, Yuriy S. Polyakov
Computer Science Department, College of Computing Sciences
New Jersey Institute of Technology, Newark, NJ 07102

Keywords: *switch location; hardware cost; concave cost function; local extrema; complexity*

Introduction

Traditionally, in continuous problems formulated for optimal switch location the cost function that is linearly proportional to the flow and the distances, expressed in terms of convex l_p norms, is considered (Love *et al.*, 1988; Verkhovsky and Polyakov, 2002; Brimberg and Chen, 1998; Brimberg *et al.*, 1998, Uster and Chen, 2000). Thus the total cost function in such a problem is convex. In reality, however, such a convex function with linear dependence on the distances is very rare. In this paper, we formulate a concave-cost switch location problem, which is more appropriate for practical applications, and provide several algorithms to solve it.

Every telecommunications link usually consists of some end equipment and many components that provide transmission media and assure good-quality transmissions. Examples of such equipment are echo suppressors in voice communications, repeaters, amplifiers, etc. Many different technologies exist to provide the above-mentioned features.

Consider a set of technologies/hardware h=1,..,z and corresponding costs:

$$C_h(d) := E_h + dR_h\,,$$

where d is the length of the transmission link, E_h is the cost of end equipment, and R_h is the cost of repeaters and other link hardware. Then the overall link cost, $c(d)$, is

$$c(d) := \min\left[E_h + dR_h\right],$$

where minimization is done over all hardware h=1,..,z. The resulting function $c(d)$ is piece-wise linear and satisfies the basic property of a concave function.

Definition: Consider $0<u<1$ and $A \le a < b \le B$. $f(x)$ is concave on an interval $[A,B]$ if and only if the following property holds: for every pair of numbers $a<b$, $f[ua+(1-u)b]<uf(a)+(1-u)f(b)$.

Traditional link-cost function

The traditional formulation of the switch location problem is (Love *et al.*, 1988):

$$\min W(S) = \sum_{i=1}^{n} w_i l_p(S, P_i) \qquad (1)$$

where $W(S)$ is the total cost; n is the number of users; w_i is the "weight" (flow) of the i-th user; $i = 1,...,n$; $P_i = (a_{i1}, a_{i2})$ is the given location of the i-th user; $S = (x_1, x_2)$ is the unknown location for the new switch. The distance between any P_i and S is given by

$$l_p(S, P_i) = [|x_1 - a_{i1}|^p + |x_2 - a_{i2}|^p]^{1/p},\ p \ge 1.$$

The total cost function is convex and many approaches exist to solve this problem (Love *et al.*, 1988).

Approximating concavity of link-cost function

To incorporate concavity in the above expression, we make both the weight and distance concave functions. In other words, we raise the weighted distance to power that is less than one. The new formulation of problem (1) becomes

$$\min W_c\left(S\right)=\sum_{i=1}^{n} w_i^{p/q} l_c\left(S,P_i\right),\ q>p\geq 2. \tag{2}$$

where

$$l_c(S,P_i)=\left[l_p(S,P_i)\right]^{p/q}=[|x_1-a_{i1}|^p+|x_2-a_{i2}|^p]^{1/q}.$$

Since $p/q<1$, it is obvious that the individual link cost function is concave of both the "weight" w_i and distance l_p. Problem (2) can be applied to the two-switch location problem studied in (Verkhovsky & Polyakov, 2003). Alternative approaches to bringing concavity to problem (1) exist. They will be studied in our future research.

Algorithms searching for optimal switch location

Three algorithms are developed and studied to solve problem (2). They are all partially based on the generalized Weiszfeld procedure (Uster and Love, 2000) developed to solve problem (1).

Direct iterative search**:** The first algorithm is a generalization of the Weiszfeld procedure to the concave case. The partial derivatives of (2) are taken and are set to zero:

$$\sum_{i=1}^{n} w_i^{p/q}\frac{1}{q}\,p\,\mathrm{sign}\left(x_k-a_{ik}\right)\left|x_k-a_{ik}\right|^{p-1}\left(\left|x_1-a_{i1}\right|^{p}+\left|x_2-a_{i1}\right|^{p}\right)^{\frac{1}{q}-1}=0,\ k=1,2;$$

After substituting $\left(x_k-a_{ik}\right)=\mathrm{sign}\left(x_k-a_{ik}\right)\left|x_k-a_{ik}\right|$ and isolating x_k, the obtained equations can be used iteratively to approach the "optimal" (the found location might not be optimal) switch location and the following formulas can be derived for the next (r+1) iterate:

$$x_k^{(r+1)}=\frac{\sum_{i=1}^{n} w_i^{p/q}a_{i1}\left|x_k-a_{ik}\right|^{p-2}/\,l_c^{q-1}\left(S,P_i\right)}{\sum_{i=1}^{n} w_i^{p/q}\left|x_k-a_{ik}\right|^{p-2}/\,l_c^{q-1}\left(S,P_i\right)},\ k=1,2; \tag{3}$$

Since the total cost function is no longer convex, Direct Iterative Search (DIS) is not guaranteed to find the optimal location due to the following reasons: 1) multiple local minimums may be present and the found one is not guaranteed to be global; 2) DIS may land at a local maximum; 3) the global minimum is on the border of the domain.

It is also obvious that expression (3) cannot be determined if one of the current coordinates coincides with a coordinate for one of the user points. To resolve this issue, exception handling routines trapping division by zero and exponentiation of zero and then restarting the procedure with slightly different initial coordinates are used. An alternative approach is to use a hyperbolic approximation (Uster and Love, 2000) wherever zero is possible.

Incremental iterative search**:** Because of the above-mentioned difficulties present in DIS, another algorithm, Incremental Iterative Search (IIS), is now developed. The hypothesis on which this algorithm is based: the global minimum gradually moves when q in problem (2) increases its value in small increments.

Let p and q be given, qt be the current value of parameter q. The algorithm starts at $qt = p$ and applies DIS, which is guaranteed to yield the optimal location when $p = 2$ and can be slightly changed to always work for $p > 2$ (Uster and Love, 2000). Once DIS finds the optimal location, qt is incremented by Δq and DIS is run again for the new value of qt. According to the above hypothesis, the global minimum should be somewhere close to the optimal location found in the previous step. Therefore, the global minimum for the new value of qt is found. Then this process repeats until qt becomes equal to q. Finally, when $qt = q$, the found optimal location should be the solution for the given p and q.

The smaller is Δq, the higher are the chances that the global minimum will be found. If the above hypothesis is true, this algorithm should always yield the global minimum unless DIS runs across a local maximum and stops there.

Incremental classical search: The same idea as used for IIS is applied to Incremental Classical Search (ICS). The only difference is that once DIS is solved for $p = q$, a different algorithm is used for finding the local minimum at each value of qt. In this case, the built-in Mathematica function *FindMinimum*, which uses various methods due to Brent: the conjugate gradient in one dimension and a modification of Powell's method in several dimensions, is applied. The *FindMinimum* is a function that has been extensively used by many Mathematica users for more than a dozen of years and thus provides a good algorithm to check the correctness of DIS for incremental steps.

Grid Search: Since all three of the above-mentioned algorithms are not guaranteed to be always true, another simple exhaustive procedure is used to always yield values close to the global minimum. The algorithm, which will be referred to as Grid Search (GS), divides the total cost function surface into a grid of mxn (in our case, 100x100). Then GS calculates the total cost for each of the nodes (points) of the grid using formula (2) and returns the coordinates for the node that gave the least total cost.

Computer experiments and their analysis

Coordinates and weights were randomly generated using the uniform distribution in Mathematica 4.1, a well-known scientific programming language developed by Wolfram Research, Inc. Weights were generated on (0,1). Coordinates were generated on (0,m), where m is the stretch factor. For all of the above calculations p = 2 was used. Sensitivity analyses on q (2..3), n (5..80), m (1..3) were conducted.

***Case* 1. *Dependence of the total cost on q*:** We ran 660 experiments considering four extreme cases: 1) n=5, m=1; 2) n=5, m=3; 3) n=80, m=1; 4) n=80, m=3. For each value of q, 30 experiments were run. Results obtained for n=5, m=3 are displayed in Table 1. For q = 2, the DIS, IIS, and CIS degenerate to the Weiszfeld procedure and thus N.A. is used for IIS and CIS.

As can be seen from the table, all three DIS-related algorithms gradually start failing when q gets larger. This fact can be easily explained from studying the total cost surface. The larger is q, the more local minimums start appearing. It is noteworthy that these new minimums occur at the user points. In a sense, user points can be considered as seeds for local minimums. As can also be seen from the table, more and more optimal locations coincide with one of the user points for higher q. This again shows that the surface gradually changes from convex to multi-modal with the users being the causes for the local minima.

Table 1. Number of times (out of 30) that the studied algorithms found the global minimum for different q.

q	Number of correct solutions for each algorithm				Number of coincidences
	DIS	IIS	CIS	GS	
2.0	30	N.A.	N.A.	30	21
2.2	29	30	30	30	20
2.4	29	29	29	30	27
2.6	26	26	26	30	29
2.8	24	25	25	30	30
3.0	24	25	25	30	30

It can be seen that in the first extreme case, with q = 2, DIS always finds the global minimum. In the other case, when q = 3, global minimum always coincides with one of the user points. Thus a simple procedure calculating the total cost at each of the user locations and selecting a point with the minimum total cost (we will refer to it as Existing User Check, EUC) can determine the global minimum in the second case.

The question that we get is why the hypothesis on which IIS and CIS were based failed to be true. The explanation is simple: new local minimums start appearing at existing user points located "far away" from the global minimum calculated for the previous values of q. Henceforth, location of the global minimum for some q is not guaranteed to be close to the one for $q+\Delta q$.

***Case* 2. *Dependence of the total cost on n*:** We ran 600 experiments considering four extreme cases: 1) q=2, m=1; 2) q=2, m=3; 3) q=3, m=1; 4) q=3, m=3. For each value of n, 30 experiments were run. Results obtained for q=3, m=3 are displayed in Table 2.

Table 2. Number of times (out of 30) that the studied algorithms found the global minimum for different n.

n	Number of correct solutions for each algorithm				Number of coincidences
	DIS	IIS	CIS	GS	
5	24	25	25	30	30
10	28	28	28	30	28
20	28	28	28	30	24
40	30	30	30	30	16
80	30	30	30	30	13

As can be seen from the table, the larger is n, the less are the chances for the DIS-based algorithms to fail. From the surface analysis it can be seen that the total cost function becomes more and more convex for higher n. When n = 80, the surface looks ideally convex. It is obvious because more users contribute to the total cost function and produce some clear overall picture. The less is n, the more local minimums the total cost function has and the more coincidences of the optimal location with one of the user points happen. Again local minimums mostly appear at the user points. Like in the previous case, all three DIS-based procedures were on the average giving the same results.

***Case* 3. *Dependence of the total cost on m*:** We ran 510 experiments considering four extreme cases: 1) q=2, n=5; 2) q=3, n=5; 3) q=2, n=80; 4) q=3, n=80. For each value of m, 30 experiments were run. Results obtained for q=3, n=5 are displayed in Table 3.

Table 3. Number of times (out of 30) that the studied algorithms found the global minimum for different m.

m	Number of correct solutions for each algorithm				Number of coincidences
	DIS	IIS	CIS	GS	
1.0	26	28	28	30	29
1.5	28	29	29	30	29
2.0	25	26	26	30	30
2.5	25	26	26	30	30
3.0	24	25	25	30	30

Although there is no clearly seen dependence that is true for all cases, larger m usually corresponds to situations when the number of coincidences between the global minimum and one of the user points becomes larger, the DIS-based procedures have higher chances to fail, and more local minimums start appearing.

Conclusions

Since in all of the above experiments, the three DIS-based algorithms demonstrated approximately the same correctness, DIS is the preferred method due to its least complexity compared to IIS and CIS. Moreover, in all of the cases when DIS failed to converge to the global minimum, one of the user points was that optimal location. Therefore, in cases where DIS might fail, both DIS and EUC need to be run, and the location corresponding to the least total cost is the optimal location. The experiments showed that this combined algorithm is necessary when n < 40 and q is in the range of 2..3. On the other hand, when n > 40 for the same range of values for q, DIS by itself correctly gives the optimal location. Since uniform random distribution was used to generate the user points for the experiments, the ranges of n and q for each of the two cases might be different if some special initial configuration is considered.

References

Brimberg, J. and Chen, R. (1998); A Note on Convergence in the Single Facility Minisum Location Problem; Computer Math. Applic., Vol. 35, No. 9 (pp. 25-31).

Brimberg, J., Chen, R. and Chen, D. (1998); Accelerating Convergence in the Fermat-Weber Location Problem; Oper. Research Letters, Vol. 22 (pp. 151-157).

Love, R. F., Morris, J. G. and G. O. Wesolowsky (1988); Facilities Location, Models and Methods; North-Holland. New York.

Uster, H. and Love, R.F. (2000); The Convergence of the Weiszfeld Algorithm; Comput. Mathem. App., Vol. 40 (pp. 443-451).

Verkhovsky, B. and Polyakov, Yu. (2002); Feedback Algorithm for the Single-Facility Minisum Problem; Annals of the European Academy of Sciences (pp. 127-136).

Verkhovsky, B. S and Polyakov, Yu. S. (2003); Highly Efficient Algorithm for Two-Switch Location Problem; Advances in Decision Technology and Intelligent Information Systems, Volume IV, K.J. Engemann and G.E. Lasker, Eds, The International Institute for Advanced Studies in Systems Research and Cybernetics (pp. 51-55).