Computational study of the step size parameter of the subgradient optimization method

Mengjie Han[1]

Abstract

The subgradient optimization method is a simple and flexible linear programming iterative algorithm. It is much simpler than Newton's method and can be applied to a wider variety of problems. It also converges when the objective function is non-differentiable. Since an efficient algorithm will not only produce a good solution but also take less computing time, we always prefer a simpler algorithm with high quality. In this study a series of step size parameters in the subgradient equation is studied. The performance is compared for a general piecewise function and a specific p-median problem. We examine how the quality of solution changes by setting five forms of step size parameter α.

Keywords: subgradient method; optimization; convex function; p-median

1 Introduction

The *subgradient optimization* method is suggested by Kiwiel (1985) and Shor (1985) for solving non-differentiable function, such as constrained linear programming. As to the ordinary gradient method, the subgradient method is extended to the non-differentiable functions. The application of subgradient method is more straightforward than other iterative methods, for example, the interior point method and the Newton method. The memory requirement is much lower due to its simplicity. This property reduces the computing burden when big data is handled.

However, the efficiency or the convergence speed of the subgradient method is likely to be affected by pre-defined parameter settings. One always would like to apply the most efficient empirical parameter settings on the specific data set. For example, the efficiency or the convergence speed has relation to the step size (a scalar on the subgradient direction) in the iteration. In this paper, the impact of the step size parameter in the subgradient equation on the convex function is studied. Specifically, an application of the subgradient method is conducted with p-median problem using Lagrangian relaxation. In this specific application, we study the impact of the step size parameter on the quality of the solutions.

Methods for solving the p-median problem are widely studied (see Reese, 2006; Mladenović, 2007; Daskin,1995). Reese (2006) summarized the literature on solution methods by surveying eight types of methods and listing 132 papers or books. Linear programming (LP) relaxation accounts for 17.4% among the 132 papers or books. Mladenović (2007) examined the metaheuristics framework for solving p-median problem. Metaheuristics has led to substantial improvement in solution quality when the problem scale is large. The Lagrangian heuristic is a specific representation of LP and metaheuristics. Daskin (1995) also showed that the Lagrangian method always give good solutions compared to constructive methods.

[1]PhD student in School of Technology and Business Studies, Dalarna Unversity, Sweden. E-mail: mea@du.se

Solving p-median problems by Lagrangian heuristics is often suggested (Beasley, 1993; Daskin, 1995; Beltran, 2004; Avella, 2012; Carrizosa, 2012). The corresponding subgradient optimization algorithm has also been suggested. A good solution can always be found by narrowing the gap between the lower bound (LB) and the best upper bound (BUB). This property provides an understanding of how good the solution is. The solution can be improved by increasing the best lower bound (BLB) and decreasing the BUB. This procedure could stop either when the critical percentage difference between LB and BUB is reached or when the parameter controlling the LB's increment becomes trivial. However, the previous studies did not examine how the LB's increment affects the quality of the solution. The LB's increment is decided by the step size parameter of the subgradient substitution. Given this open question, the aim of this paper is to examine how the step size parameter in the subgradient equation affect the performance of a convex function through a general piecewise example and several specific p-median problems.

The remaining parts of this paper are sectionally organized in subgradient method and the impact of step size, p-median problem, computational results and conclusions.

2 Subgradient method and the impact of step size

The subgradient method provides a framework of minimizing a convex function $f : \boldsymbol{R}^n \rightarrow \boldsymbol{R}$ by using the iterative equation:

$$x^{(k+1)} = x^{(k)} - \alpha(k) g^{(k)}. \tag{2.1}$$

In (2.1) $x^{(k)}$ is the kth iteration of the argument x of the function. $g^{(k)}$ is an arbitrary subgradient of f at $x^{(k)}$. $\alpha(k)$ is the step size. The convergence of (2.1) can be proved by Shor (1985).

2.1 step size forms

Five typical rules of step size are listed in Stephen and Almir (2008). They can be summarized as:

- constant size: $\alpha(k) = \xi$
- constant step length: $\alpha(k) = \xi / \|g^{(k)}\|_2$
- square summable but not summable: $\alpha(k) = \xi/(b+k)$
- nonsummable diminishing: $\alpha(k) = \xi/\sqrt{k}$
- nomsummable diminishing step length: $\alpha(k) = \xi(k)/\|g^{(k)}\|_2$

The form of the step size is pre-set and will not change. The top two forms, $\alpha(k) = \xi$ and $\alpha(k) = \xi/\|g^{(k)}\|_2$, are not examined since they are constant size or step length which are lack of variation for p-median problem. On the other hand, the bottom three forms are studied. We restrict $\xi(k)$ such that $\xi(k)/\|g^{(k)}\|_2$ can be represented by an exponential decreasing function of $\alpha(k)$. Thus, we study $\alpha(k) = \xi/k$, $\alpha(k) = \xi/\sqrt{k}$, $\alpha(k) = \xi/1.05^k$, $\alpha(k) = \xi/2^k$ and $\alpha(k) = \xi/\exp(k)$ in this paper. We first examine the step size impact on a general piecewise convex function and then on the p-median problem.

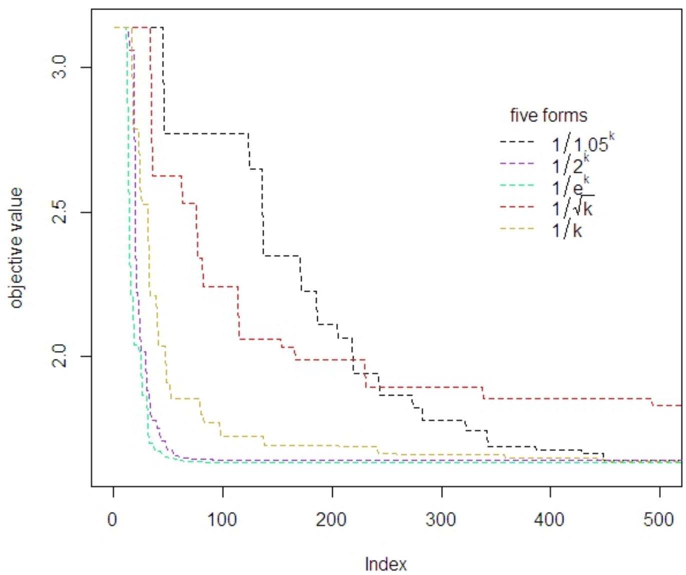

Figure 1: Objective values of picewise convex function when five forms of step sizes are compared

2.2 general impact on convex function

We consider to minimize the function:

$$f(x) = \max_i(\boldsymbol{a}_i^T x + b_i)$$

where $x \in \boldsymbol{R}^n$ and a subgradient $\boldsymbol{g}$ can be taken as $\boldsymbol{g} = \nabla f(x) = \boldsymbol{a}_j$ ($\boldsymbol{a}_j^T x + b_j$ maximizes $\boldsymbol{a}_i^T x + b_i$, $i = 1, \dots, m$). In our experiment, we take $m = 100$ and the dimension of x being 10. Both $\boldsymbol{a} \sim MVN(\boldsymbol{0}, \boldsymbol{I})$ and $b \sim N(0, 1)$. The initial value of constant ξ is 1. The initial value of $\boldsymbol{x}$ is $\boldsymbol{0}$. We run the subgradient iteration 1000 times. Figure 1 shows a non-increased objective values of the function against the number of iterations. The objective value is taken when there is a improvement of the objective function. Otherwise, it is taken as the minimum value in the previous iterations.

In Figure 1, $\alpha(k) = 1/2^k$ and $\alpha(k) = 1/\exp(k)$ have similar converging patterns and quickly approach the "optimal" bottom. The convergence speed of $\alpha(k) = 1/\sqrt{k}$ is a bit slower, but it has a steep slop before 100 iterations as well. $\alpha(k) = 1/\sqrt{k}$ does not have a fast improvement after 200 iterations, while $\alpha(k) = 1/1.05^k$ has an approximately uniform convergence speed. However, $\alpha(k) = 1/\sqrt{k}$ is still far away from the "optimal" bottom. In short, $\alpha(k) = 1/2^k$ and $\alpha(k) = 1/\exp(k)$ provide uniformly good solutions which would is more efficient when dealing with big data.

3 *p*-median problem

An important application of the subgradient method is solving p-median problems. Here, the p-median problem is formulated by integer linear programming. It is defined as follows.

$$\text{Minimize:} \quad \sum_i \sum_j h_i d_{ij} Y_{ij} \tag{3.1}$$

subject to:

$$\sum_j Y_{ij} = 1 \quad \forall i \tag{3.2}$$

$$\sum_j X_j = P \tag{3.3}$$

$$Y_{ij} - X_j \leqslant 0 \quad \forall i, j \tag{3.4}$$

$$X_j = 0, 1 \quad \forall j \tag{3.5}$$

$$Y_{i,j} = 0, 1 \quad \forall i, j \tag{3.6}$$

In (3.1), h_i is the weight on each demand point and d_{ij} is the cost of the edge. Y_{ij} is the decision variable indicating that if a trip between node i and j is made or not. Constraint (3.2) ensures that every demand point must be assigned to one facility. In (3.3) X_j is decision variable and it ensures that the number of facilities to be located is P. Constraint (3.4) indicates that no demand point i is assigned to j unless there is a facility. In constraint (3.5) and (3.6) the value 1 means that the locating (X) or travelling (Y) decision is made. 0 means that the decision is not made.

To solve this problem using sugradient method, the Lagrangian relaxation must be made. Since the number of facilities, P, is fixed, we cannot relax the locating decision variable X_j. Consequently, the relaxation is necessarily put on the travelling decision variable Y_{ij}. It could be made either on (3.2) or on (3.4). In this paper, we only consider the case for (3.2), because the same procedure would be applied on (3.4). We do not repeat this for (3.4). What we need to do is to relax this problem for fixed values of the Lagrange multipliers, find primal feasible solutions from the relaxed solution and improve the Lagrange multipliers (Daskin, 1995). Consider relaxing constraint (3.2), we have

$$\begin{aligned} \text{Minimize:} \quad & \sum_i \sum_j h_i d_{ij} Y_{ij} + \sum_i \lambda_i (1 - \sum_j Y_{ij}) \\ = & \sum_i \sum_j (h_i d_{ij} - \lambda_i) Y_{ij} + \sum_i \lambda_i \end{aligned} \tag{3.7}$$

with constraints (3.3)–(3.6) unchanged. In order to minimize the objective function for fixed values of λ_i, we set $Y_{ij} = 1$ when $h_i d_{ij} - \lambda_i < 0$ and $Y_{ij} = 0$ otherwise. The corresponding value of X_j is 1. A set of initial values of λ_is are given by the mean weighted distance between each node and the demand points.

Lagrange multipliers are updated in each iteration. The step size value in the kth iteration for the multipliers $T^{(k)}$ is :

$$T^{(k)} = \frac{\alpha^{(k)}(BUB - \mathcal{L}^{(k)})}{\sum_i \{\sum_j Y_{ij}^{(k)} - 1\}^2}, \tag{3.8}$$

where $T^{(k)}$ is the kth step size value; BUB is the minimum upper bound of the objective function until the kth iteration; $\mathcal{L}$ is the value evaluated by (3.7) at the kth iteration; $\sum_j Y_{ij}^{(k)}$ is the current optimal value of the decision variable. The Lagrangian multipliers, λ_i, are updated by:

$$\lambda_i^{(k+1)} = \max\{0, \lambda_i^{(k)} - T^{(k)}(\sum_j Y_{ij}^{(k)} - 1)\}. \tag{3.9}$$

A general working scheme is:

- step 1 Plug the initial values or updated values of λ_i into (3.7) and identify the p medians according to $h_i d_{ij} - \lambda_i$;
- step 2 According to p medians in step 1, evaluate the subgradient $g^{(k)} = 1 - \sum_j Y_{ij}$, BUB and $\mathcal{L}^{(k)}$. If the stopping criteria is met, stop. Otherwise, go to step 3;
- step 3 Update $T^{(k)}$ using (3.8);
- step 4 Update Lagrangian multipliers $\lambda_i^{(k)}$s using (3.9). Then go to step 1 with new λ_is.

The lower bound (LB) in each iteration is decided by the value of λ_is. The step size $T^{(k)}$ will affect the update speed of λ_is. It goes to 0 when the number of iterations tend to infinity. When it goes slowly, the increment of LB would be fast but unstable. This leads to inaccurate estimates of the LB. On the other hand, when the update speed goes too fast, the update of LB is slow. The non-update would easily happen such that the difference between BUB and BLB remains even though more iterations are made. The danger will arise if the inappropriate step size is computed. Thus, a good choice of executed parameter controlling the update speed would make the algorithm more efficient.

4 Computational results

In this section, we study the parameter, α, controlling the step size. Daskin (1995) suggested an initial value of 2 and a halved decreasing factor after 5 failures of changing; Avella (2012) suggested an initial value of 1.5 and a 1.01 decreasing factor after one failure of changing. We could also consider other alternative initial values instead of that in the previous studies. However, that is only a minor issue and not related to the step size function. Thus, we skip the analysis of the initial values.

The complexity in our study is different from Daskin (1995) and Avella (2012). We take medium sized problems from the OR-library (Beasley, 1990). The OR-library consists of 40 test p-median problems. The optimal solutions are given. We pick eight cases. N varies from 100 to 900 and P varies from 5 to 80. A subset is picked in our study by only selecting two cases for each $N = 100, 200, 400, 800$. The parameter α take five forms. Following

Table 1: Lagrangian settings testing a subset of OR-library

$\alpha(k)$	form 1: ξ/k form 2: $\xi/\sqrt{k}$ form 3: $\xi/1.05^k$ form 4: $\xi/2^k$ form 5: $\xi/\exp(k)$
n (number of failures before changing α)	5
restart the counter when α changed	Yes
critical difference	0.01
initial values of λ_is	$\sum_j h_i d_{ij} Y_{ij} / \sum j$
maximum iterations after no improvement on BUB	$m = 1000$ and $m = 100$

Stephen and Almir (2008), we take the forms of $\alpha(k) = \xi/k$, $\alpha(k) = \xi/\sqrt{k}$, $\alpha(k) = \xi/1.05^k$, $\alpha(k) = \xi/2^k$ and $\alpha(k) = \xi/\exp(k)$.The procedure settings are shown in Table 1.

In Table 1, $\alpha(k)$ is the step size function of. We take $\xi = 1$ as we did for the piecewise function $f(x)$. n is a counter recording the number of 5 consecutive failures. As suggested by Daskin (1995), we do not further elaborate the impact from the counter. The critical difference takes the value of 1% of the optimal solution. This is only a criterion for known optimal values and it can be largely affected by the type of the problem. Considering that, the algorithm also stops if no improvement of BUB is found after preset number of iterations. Here we compare 100 and 1,000. Given the settings, the results are shown in Table 2 and Table 3.

In Table 2 and Table 3, optimal solution values are given for two stopping criteria. The problem complexity varies. We compare different forms of $\alpha(k)$. BLBs (best lower bound), BUBs (best upper bound), deviations ($\frac{\text{BUB}-\text{Optimal}}{\text{Optimal}} \times 100\%$), U/L ($\frac{\text{BUB}}{\text{BLB}}$) and the number of iterations. The optimal BUB and U/L are marked in bold.

Table 2 shows the solutions for $m = 100$. For pmed 1 and pmed 6 of the OR-library, the exact optimal solutions are obtained. For pemd 35, an almost exact solution is also obtained. On the other hand, for pmed 4, pmed 9, pmed 18 and pmed 37, the BLB is much closer to the optimal. For most of the cases, the step size function with the minimum U/L ratio gives the lowest BUB. It is an indication of the good quality of the algorithm even though $1/1.05^k$ performs very bad in pmed 18 and pmed 37. It is no surprise that more exact solutions appear when the number of iterations is increased, for example, $1/1.05^k$ in pmed 1 and pmed 6; $1/k$ in pmed 4 and pmed 35 in Table 3. Similarly, we also improve the quality of BLBs. The worst deviation is 17.70 for $m = 1000$ instead of 44.14 for $m = 100$.

There are several overall tendencies we can draw from Table 2 and Table 3. Firstly, $1/2^k$ and $1/\exp(k)$ are relatively stable which is also in accordance with piecewise function we studied before. This can be seen not only for less complicated problem but also for the complicated case. However, there is no obvious tendency of which one will dominates. Secondly, it is

Table 2: Comparison of optimal solutions for different step size decreasing speed ($m = 100$)

File No.	$f_n(\alpha)$	BLB	BUB	Optimal	Deviation (%)	U/L	Iterations
pmed 1 ($N = 100$ $P = 5$)	$1/k$	5803	5821	5819	0.03	1.003	65
	$1/\sqrt{k}$	5811	5821	5819	0.03	**1.002**	98
	$1/1.05^k$	5521	6455	5819	10.93	1.169	103
	$1/2^k$	5796	**5819**	5819	0.00	1.005	46
	$1/exp(k)$	5796	5821	5819	0.03	1.005	53
pmed 4 ($N = 100$ $P = 20$)	$1/k$	3032	3265	3034	7.61	1.077	300
	$1/\sqrt{k}$	3030	3297	3034	8.67	1.088	435
	$1/1.05^k$	3034	**3182**	3034	4.88	**1.049**	535
	$1/2^k$	3034	**3182**	3034	4.88	**1.049**	249
	$1/exp(k)$	3034	**3182**	3034	4.88	**1.049**	164
pmed 6 ($N = 200$ $P = 5$)	$1/k$	7770	8238	7824	5.29	1.060	143
	$1/\sqrt{k}$	7760	8195	7824	4.74	1.056	202
	$1/1.05^k$	7459	8948	7824	14.37	1.200	153
	$1/2^k$	7753	**7824**	7824	0.00	**1.009**	145
	$1/exp(k)$	7751	**7824**	7824	0.00	**1.009**	56
pmed 9 ($N = 200$ $P = 40$)	$1/k$	2732	**3051**	2734	11.59	**1.117**	471
	$1/\sqrt{k}$	2719	3264	2734	20.04	1.200	386
	$1/1.05^k$	2725	3239	2734	18.47	1.189	451
	$1/2^k$	2732	3069	2734	12.25	1.123	282
	$1/exp(k)$	2732	3127	2734	14.37	1.145	297
pmed 16 ($N = 400$ $P = 5$)	$1/k$	8090	8253	8162	1.411	1.020	231
	$1/\sqrt{k}$	8086	8240	8162	0.96	1.019	261
	$1/1.05^k$	8092	**8185**	8162	0.28	**1.011**	534
	$1/2^k$	8088	8239	8162	0.94	1.019	210
	$1/exp(k)$	8080	8206	8162	0.54	1.016	156
pmed 18 ($N = 400$ $P = 40$)	$1/k$	4807	5021	4809	4.22	1.043	256
	$1/\sqrt{k}$	4801	5150	4809	7.09	1.073	516
	$1/1.05^k$	3848	6913	4809	43.75	1.797	101
	$1/2^k$	4805	**4865**	4809	1.16	**1.012**	216
	$1/exp(k)$	4803	4902	4809	1.93	1.021	269
pmed 35 ($N = 800$ $P = 5$)	$1/k$	10288	10504	10400	0.01	1.021	124
	$1/\sqrt{k}$	10296	**10401**	10400	0.01	**1.010**	254
	$1/1.05^k$	10183	10710	10400	2.98	1.052	144
	$1/2^k$	10286	**10401**	10400	0.01	1.011	201
	$1/exp(k)$	10282	**10401**	10400	0.01	1.012	239
pmed 37 ($N = 800$ $P = 80$)	$1/k$	5056	5248	5057	3.78	1.038	306
	$1/\sqrt{k}$	5033	5577	5057	10.28	1.108	342
	$1/1.05^k$	3820	7289	5057	44.14	1.908	101
	$1/2^k$	5055	5137	5057	1.58	1.016	314
	$1/exp(k)$	5051	**5100**	5057	0.85	**1.010**	161

Table 3: Comparison of optimal solutions for different step size decreasing speed ($m = 1000$)

File No.	$\alpha(k)$	BLB	BUB	Optimal	Deviation (%)	U/L	Iterations
pmed 1 ($N = 100$ $P = 5$)	$1/k$	5804	5821	5819	0.03	1.003	65
	$1/\sqrt{k}$	5811	5821	5819	0.03	1.002	98
	$1/1.05^k$	5815	**5819**	5819	0.00	**1.001**	239
	$1/2^k$	5796	**5819**	5819	0.00	1.004	46
	$1/exp(k)$	5796	5821	5819	0.03	1.004	53
pmed 4 ($N = 100$ $P = 20$)	$1/k$	3034	**3182**	3034	4.88	**1.049**	1975
	$1/\sqrt{k}$	3031	3259	3034	7.42	1.075	1580
	$1/1.05^k$	3034	**3182**	3034	4.88	**1.049**	1435
	$1/2^k$	3034	**3182**	3034	4.88	**1.049**	1162
	$1/exp(k)$	3034	**3182**	3034	4.88	**1.049**	1064
pmed 6 ($N = 200$ $P = 5$)	$1/k$	7782	8086	7824	3.35	1.039	1417
	$1/\sqrt{k}$	7783	7867	7824	0.66	1.011	1853
	$1/1.05^k$	7783	**7824**	7824	0.00	**1.005**	698
	$1/2^k$	7753	**7824**	7824	0.00	1.009	145
	$1/exp(k)$	7751	**7824**	7824	0.00	1.009	56
pmed 9 ($N = 200$ $P = 40$)	$1/k$	2733	**3051**	2734	11.59	**1.116**	1371
	$1/\sqrt{k}$	2720	3217	2734	17.70	1.183	1400
	$1/1.05^k$	2734	3098	2734	13.31	1.133	1674
	$1/2^k$	2732	3069	2734	12.25	1.123	1182
	$1/exp(k)$	2732	3073	2734	12.40	1.125	1359
pmed 16 ($N = 400$ $P = 5$)	$1/k$	8091	8219	8162	0.70	1.016	1685
	$1/\sqrt{k}$	8088	8240	8162	0.96	1.019	1161
	$1/1.05^k$	8092	**8162**	8162	0.00	**1.009**	859
	$1/2^k$	8088	8183	8162	0.26	1.012	1433
	$1/exp(k)$	8080	8206	8162	0.54	1.016	1056
pmed 18 ($N = 400$ $P = 40$)	$1/k$	4808	4943	4809	2.79	1.028	1499
	$1/\sqrt{k}$	4807	4957	4809	3.08	1.031	3453
	$1/1.05^k$	4809	4894	4809	1.77	1.018	2707
	$1/2^k$	4805	**4841**	4809	0.67	**1.007**	314
	$1/exp(k)$	4803	4877	4809	1.41	1.015	1726
pmed 35 ($N = 800$ $P = 5$)	$1/k$	10297	**10401**	10400	0.01	**1.010**	1453
	$1/\sqrt{k}$	10297	**10401**	10400	0.01	**1.010**	348
	$\alpha/1.05^k$	10302	10481	10400	0.78	1.017	1696
	$\alpha/2^k$	10286	**10401**	10400	0.01	1.011	1011
	$\alpha/exp(k)$	10282	**10401**	10400	0.01	1.012	1139
pmed 37 ($N = 800$ $P = 80$)	$1/k$	5057	5124	5057	1.32	1.013	1779
	$1/\sqrt{k}$	5056	5201	5057	2.85	1.029	3281
	$1/1.05^k$	5057	5140	5057	1.64	1.016	2159
	$1/2^k$	5055	5123	5057	1.31	1.013	2009
	$1/exp(k)$	5051	**5100**	5057	0.85	**1.010**	161

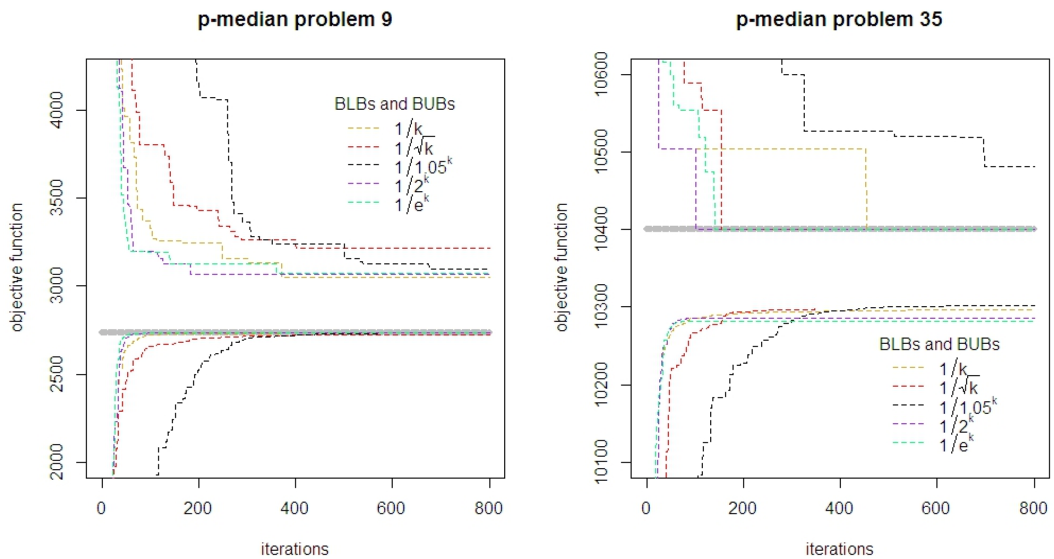

Figure 2: Changes for BLB and BUB for file No.9 and No.35

difficult for $1/\sqrt{k}$ to perform better that the rest of 4 forms to have a optimal BUB, which is in accordance with the piecewise function. One reason is that when the number of iterations is large, a slightly short step size is required. Too large steps can bring infeasible solutions, which to some extent enlarge the gaps between BLBs and BUBs. Thirdly, $1/k$ and $1/1.05^k$ are too sensitive to the stopping criterion, which is not seen in the general piece wise function. Decision for stop the algorithm should be very carefully made. One suggested way is to visualize the convergence curve and to terminate the iteration when the curve becomes flat.

Generally speaking, the BLB and the BUB tend to complement each other. In other words, one can always make an inference that either the BLB or the BUB would be the benchmark when there is a gap between BLB and BUB. In Figure 2, for example, two extreme cases are shown. The grey line represents the optimal value. The left panel shows the first 800 objective values for five forms of step size functions in problem 9 (pmed 9). The right one shows the values in problem 35 (pmed 35). For pmed 9, the BLBs quickly converge to the optimal. However, only sub-optimal BUBs are obtained. On the contrary, pmed 35 has good BUBs and bad BLBs. Thus, either the BLB or BUB is likely to reach the sub-optimal. When this happens, a complement algorithm could be involved to improve the solution.

5 Conclusion

In this paper, we studied how the decreasing speed of step size in the subgradient optimization method affects the performance of the convergence. The subgradient optimization method is simpler in solving linear programming. However, the choice of the step function in the subgradient equation can bring uncertainties to the solution. Thus, we conduct our study on examining how the step size function parameter α affect the performance. Both a general piecewise function and a specific p-median problem are studied. The p-median problem is represented by linear programming and the corresponding Lagragian relaxation is added.

We examined five forms of the step size parameters α. One is square summable but not summable form $\alpha(k) = \xi/(b+k)$. One is nonsummable diminishing form $\alpha(k) = \xi/\sqrt{k}$. Three are nonsummable diminishing step length forms $\alpha(k) = \xi/1.05^k$, $\alpha(k) = \xi/2^k$ and $\alpha(k) = \xi/\exp(k)$. We evaluated the best upper bound, best lower bound, and the required iterations to reach our stopping criteria. We have the following conclusions.

Firstly, the nonsummable diminishing step size function $\alpha(k) = \xi/\sqrt{k}$ has its limitation when the number of iterations are large. For both the general piecewise function and the p-median problem, nonsummable diminishing step size function performs bad and easily goes into the suboptimal solution. Two nonsummable diminishing step length function $\alpha(k) = \xi/2^k$ and $\alpha(k) = \xi/\exp(k)$ have similar behaviors and stable solutions. As long as the problem is not likely to lead to the suboptimal solutions, step size function $\alpha(k) = \xi/2^k$ and $\alpha(k) = \xi/\exp(k)$ always give fast convergence for both BLB and BUB. This is found both in general piecewise function and p-median problems. The square summable but not summable form $\alpha(k) = \xi/(b+k)$ as well as nonsummable diminishing form $\alpha(k) = \xi/1.05^k$ are unstable. They are also sensitive to the number of iterations.

Secondly, from our empirical result, the quality of the solution will be largely affected by the specific type of the problem. The problem characteristic may have influence on the difficulties of avoiding suboptimal solutions. If it is easy to avoid suboptimal solutions for a specific step size function α, one can make a good inference. On the other hand, if the subgradient method can always produce the suboptimal solution, a complement algorithm can be considered to get out from the suboptimal.

Thirdly, the problem complexity has little impact. We cannot assert that good solutions can be found for a less complex problem and bad solutions for a complex solution for a subgradient method.

References

[1] Avella, P., Boccia, M., Salerno, S. and Vasilyev, I., 2012. An aggregation heuristic for large p-median problem, *Computers and Operations Research*, 32, 1625–1632.

[2] Beasley, J.E., 1990. OR Library: distributing test problems by electronic mail, *Journal of the Operational Research*, 41(11), 1069-1072.

[3] Beasley, J.E., 1993. Lagrangian heuristics for location problems, *European Journal of Operational Research*, 65, 383–399.

[4] Beltran, C., Tadonki, C. and Vial, J.-Ph., 2004, Solving the p-median problem with a semi-lagrangian relaxation, *Logilab* Report, HEC, University of Geneva, Switzerland.

[5] Carrizosa, E., Ushakov, A. and Vasilyev, I., 2012. A computational study of a nonlinear minsum facility location problem, *Computers and Operations Research*, 32, 2625–2633.

[6] Daskin, M., 1995, *Network and discrete location*, Wiley, New York.

[7] Kiwiel, K.C., 1985, Methods of decent for nondifferentiable optimization. Springer Verlag, Berlin.

[8] Mladenović, N., Brimberg, J., Hansen, P. and Moreno-Pérez, JA., 2007. The p-median problem: a survey of mataheuristic approaches, *European Journal of Operational Research*, 179(3), 927–939.

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

[10] Shor, N.Z., 1985, Minimization methods for non-differentiable functions. Springer Verlag, New York.

[11] Stephen, B. and Almir, M., 2008, Notes for EE364b, Stanford University, Winter 2006–2007.