*April 28th, 2018*

**Review of some optimization algorithms**

The focus of the discussion is on the ***mechanics of the algorithms*** and, therefore, proofs of convergence are not given explicitly. The ***convex combinations method*** mentioned in the last section is the basis for solving many equilibrium problems.

**4.1 One-dimensional minimization**

This section deals with the minimization of a nonlinear function of a single variable, . Assumption that:  
 (1)lies within some finite interval  and that is continuous and uniquely defined everywhere in that interval.(regularity conditions 正则条件, as mentioned in chapter 2)

(2) is ditonic over the interval .

The algorithms used for the single dimensional minimization are shown in the Figure 4.1.1:

Figure 4.1.1 The algorithms used for the single dimensional minimization

**Interval reduction methods**

The steps of interval reduction methods are as follows:  
**Step 1** designating the first current interval ;

**Step 2** examine this interval. If the ***interval is less than a predetermined constant***, then execute step 3; if not, divide it into two parts: the part in which the minimum cannot lie and the current interval for the next iteration. The part in which the minimum cannot lie is discarded and the procedure is repeated for the new current interval.

**Step3** the estimate of is the midpoint, , of the interval remaining after  iterations, , that is



The current interval in the iteration is a portion of , denoted ,which was determined to include the minimum point, . The size of the interval is denoted by . The interval reduction ratio for the iteration is calculated as:



Usually, the reduction ratio is a constant.

If the optimum has to be estimated with a tolerance of (i.e. must lie within ), then the number of iterations（the number of the objective function was calculated） can be calculated as a function of the length, .



That is

[4.1]

Where  means the integer part of the argument.

The various interval reduction algorithms differ from each other only in the *rules used to examine the current interval* and to *decide which portion of which can be discarded*.

**Golden section method.** Firstly the interval discarding process isexplained. Secondly, the choice rule for selecting the interior points is introduced, and it is the unique feature of this method.

The discarding mechanism is demonstrated in Figure 4.1.2, depicting a ditonic function ,  ,which has to be minimized in the interval . The steps of the algorithm are as follows at the iteration:

Step 1 select two interior points and ,.

Step 2 since , the interval  can be discarded.

Step 3 the new current interval (for the iteration) is where ,.

The interval reduction process continues with two new interior points, and . Attention, the golden section procedure make use of one of the interior points from the last interval ( where the function value is already known).

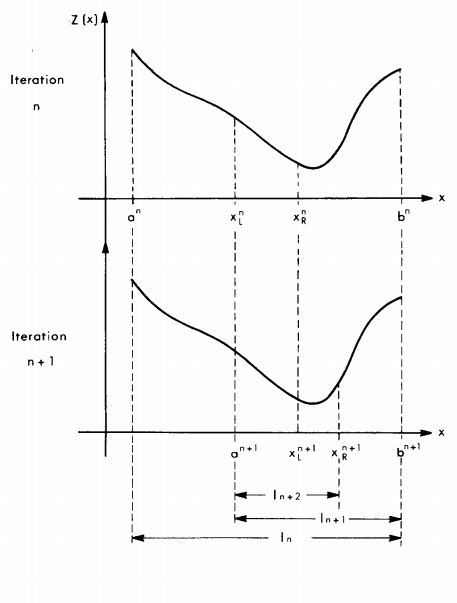


Figure 4.1.2 the interval reduction sequence followed by the golden section method

The interval reduction ratio is a constant,  (0.618),which is kwon as the "golden section method". Such a sequence of intervals leads to a situation in which



The flowchart of the algorithm is presented in Figure 4.1.3.

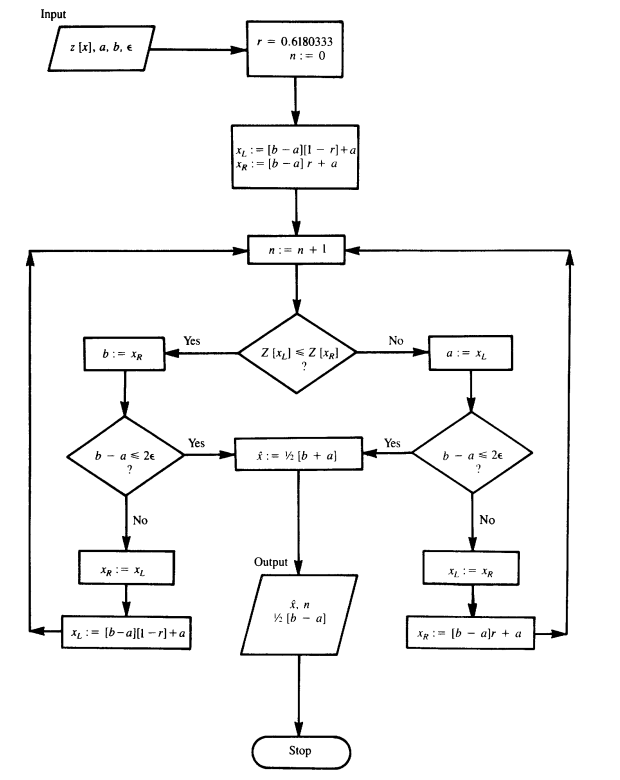


Figure 4.1.3 Flowchart of the golden section algorithm

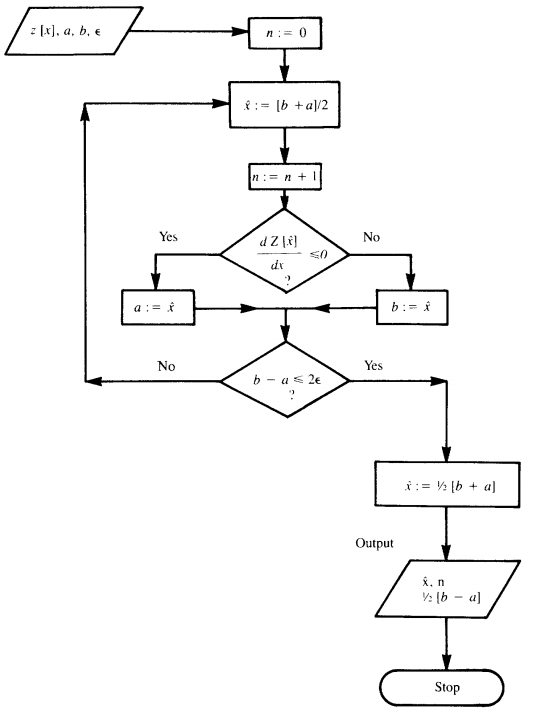
**Bisection method.** The function to be minimized by this method must satisfy the following two conditions:

(1) The derivative of the function can be evaluated easily. (the reason why using  directly may be is that the equality is difficult to solve, while can be obtained easily given ).

(2) ditonic function is monotonic on each side of the minimum.(单峰函数本来就是这样的啊). In other words, the derivative of the function to be minimized, , is negative for and positive for .(the function can be minimized by golden section method also mast satisfy this attribute).

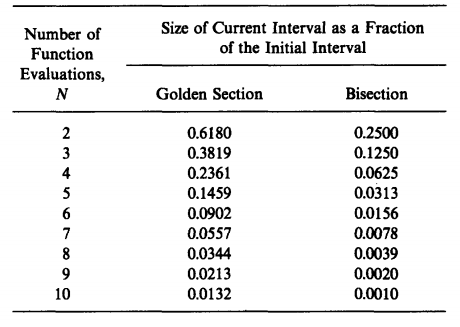
At the iteration, the current interval is , the midpoint of which is . If , then, meaning that the interval should be discarded. If , then, the search can focus on the interval . The iteration process terminate when the ***convergence criterion*** is met. The reduction ratio for this method is . The number of iterations required for a given accuracy can be determined by Eq.[4.1].

Figure 4.1.4 depicts a flowchart of the bisection method.



The table 4.1 demonstrates that the convergence rate of the bisection method is almost twice that of the golden section method. Because, the bisection method discards the half of the original interval while the golden section method only discards 0.382 of the original interval. However, the bisection method requires that the derivative of the function to be minimized should be evaluated in every iteration.

Table 4.1 Convergence rate for interval reduction methods (versus for each algorithm)



**Curve-Fitting Methods**

The function to be minimized is required to be not only ***ditonic*** but also relatively ***smooth.*** Curve-fitting methods work by iteratively improving a ***current solution point.*** All the methods described in this section use a parabola to fit the original function. These algorithms differ from each other in the technique used to generate the approximation of the objective function.

Curve-fitting methods generate a series of points , that converge to the minimum, . The algorithm terminates when a convergence criterion is met. The convergence criterion for curve-fitting methods can be based on the following rules:

(1) The marginal contribution of an iteration to the improvement in the solution becomes small, that is



Where is a predetermined constant.

(2) A dimensionless constant based on the relative change between successive solutions can be used to test for the convergence. For example



(3) The algorithms can terminate on the basis of the change in the variable value,



(4) If the derivative of  at  is close to zero, the algorithm terminates.

**Newton's Method.** Newton's method approximates at each iteration with a quadratic fit at the current point, . The fitted curve, ,is given by



The next solution, , is located at the point that minimizes . That is

****

Thus  can be obtained by the following formula

[4.4b]

**False position Method.** When the second derivative are unavailable or difficult to evaluate in Eq.[4.4b], the false position method is used. The following expression can be substituted for the second derivative in Eq.[4.4b]

****

**Quadratic fit Method.** The quadratic fit is based on three points, ,and . At the iteration



The set of three points used in the next iteration includes , and the two points out of ,and  with the smallest value of . These points are labeled ,and  and the procedure is repeated.

**Summary.** The choice among the various curve-fitting methods should be guided by the difficulty of calculating derivatives. The concrete rules are shown in the table

|  |  |  |  |
| --- | --- | --- | --- |
| Whether the derivative can be calculated easily | The first derivative | The second derivative | The choice result |
| The first condition | √ | √ | Newton's method |
| The second condition | √ | × | False position method |
| The third condition | × | × | Quadratic fit method |

**4.2 Multivariable minimization**

This section deals with the minimization principles of nonlinear convex program of several variables. The algorithms included here are all ***descent methods***. In each case, the core of any algorithmic procedure focuses on generating a point from , so that ,

[4.7]

Where is a descent direction, is the nonlinear scalar known as the move size (or "step size").

The relationship between gradient and descent direction can be written as, which can be shown as the Figure 4.2.1.

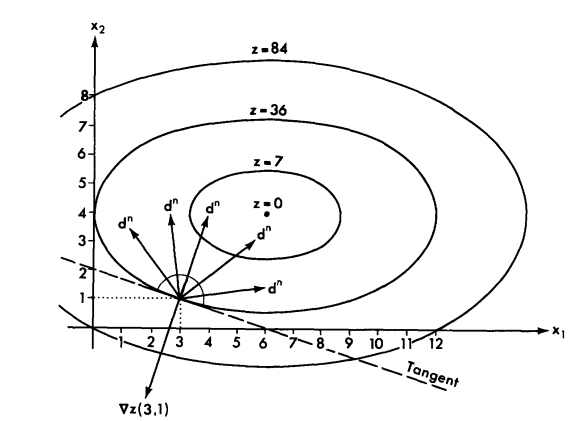


Figure 4.2.1 Contour lines of 

All the descent algorithm differ mostly in the rule used for finding the descent direction. The move size is chosen so that the objective function is minimized along the descent direction.

For the unconstrained minimization, the algorithmic iteration can be used to summarize all descent methods. For the constrained minimization, the algorithms should guarantee the descent direction and move size is feasible.

**Unconstrained minimization algorithms**

The convergence criteria used to terminate the algorithmic iterations are as follows:  
(1)The marginal contribution of successive iterations is less than the predetermined constant:  
[4.8a]

(2) The algorithm can be terminated if the elements of the gradient vector are close to zero"  
[4.8b]

(3) Change in the variables between successive iterations is less than the predetermined constant:  
[4.8c]

Or [4.8d]

In these criteria, is a predetermined tolerance (different for each criterion) selected especially for each problem on the desired degree of accuracy.

**The method of steepest descent**. For the method of steepest descent, the direction of search is opposite to the gradient direction, which means that each move is made in the direction in which  decreases the most. The length of the move is determined by the points where the value of stops decreasing and starts increasing. Therefore, the basic iteration is given by  
[4.9]

Where,  is the optimal move.

The gradient can be found either analytically or, when the derivatives are difficult to evaluate, by using numerical methods. For the analytical method, . For the latter methods, can be computed numerically by approximating the partial derivative of with respect to each . In other words:  


Where  is a small interval associated with the component of .

In order to find the value of , the function

[4.11]

has to be minimized with respect to subject to the constraint that . The value of that minimizes the Eq.[4.11] is . Finding is a one-dimensional minimization problem that can be solved by using any of the techniques mentioned in section 4.1 except for the interval reduction methods, since the problem here is unconstrained and therefore no initial interval exists. If the problem is simple (and convex), can be determined analytically by solving



Figure 4.2.2 depicts a typical sequence of solutions generated by the steepest descent algorithm. For the successive search direction:  


Due to the optimization in the line search. If the contours are circular, the steepest descent method converges in a single step. For the general quadratic form, the "zigzagging" becomes more pronounced, and the zigzagging means that the steepest descent method requires a large number of iterations.

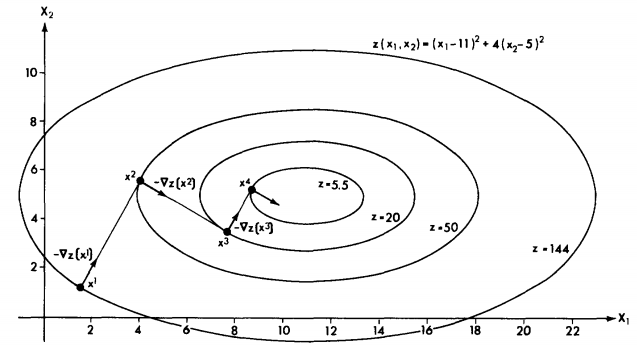
****

Figure 4.2.2 Convergence pattern of the steepest descent algorithm

**Other methods. Attention :**the problems arising in the course of studying urban network are typically large (i.e. including many variables), the methods referring to the calculation of the Hessian or even the approximations of Hessian are precluded. The following methods introduced are those methods which are precluded for solving the urban network problems.

For Newton's method, the objective function is approximated by a second-order Taylor series. This approximation then is minimized and the new solution is taken to be the point that minimizes the approximation. The resulting algorithmic step is given by

[4.2]

This method can be expressed in the standard for [4.7a] by defining , and , .

Newton's method is more efficient than the steepest descent method because its search direction uses more information about the shape of at . The approach underlying quasi-Newton method is try to construct an approximation to the inverse Hessian.

**Constrained minimization algorithms**

The study of transportation network equilibrium problems involves equivalent mathematical programs which include extensively ***linear constraints***. The discussion of constrained minimization limited, therefore, to these types of constraints. The techniques reviewed in this section are applicable both ***equality and inequality*** constraints.

The basic framework outlined in the preceding section for unconstrained minimization (i.e. finding a descent direction and advancing by a optimal amount along this direction ) can be used to describe the algorithms for constrained optimization as well. The added difficulty in constrained minimization is to ***maintain feasibility***. In other words:  
 (1) the ***search direction*** has to point toward a region where some feasible solutions lie.

(2) the ***search for optimal step size*** has to be constrained to feasible points only.

The focus of this section is on these topics, which are common to all ***feasible direction*** methods.

The convergence criteria can be similar to the ones used for constrained minimization (see Eq. [4.8]) except the gradient-based convergence criteria (Eq. [4.8b]), since the minimum of a constrained program is not necessarily a stationary point of the objective function.

**Maintaining feasibility in a given direction.** Consider thestandard form of a minimization program (see Chapter 2)

[4.13a]

Subject to [4.13b]

Where the constraints are all linear (the nonnegative constraints are included in Eqs.[4.13]).

Assume further that the current solution is , a point that may lie on the boundary of the feasible region. In this case, some of the constraints are binding, that is,



Where is the index set of the binding constraints at the  iteration.

The other constraints are nonbinding, that is



Once a feasible descent direction, , is obtained, the maximum move size that can be taken without violating any constraints should be determined. Only the nonbinding constraints should be considered. Because the binding constraints determine the descent direction  and the start point, however the other constraints (nonbinding constraints ) determine how long the move is along the descent direction , because the next iteration point should be within the feasible region.

For  to be feasible , the nonbinding constraints must not be violated by the new solution, . In other words, the following must hold:



Since the current solution is feasible, that is , then whether the new solution, , is feasible is determined by the .

For a given constraint , if , the inequality is always satisfied .

However, if , there are some limitation for the value of . The above inequality can be written as:



That is,

[4.15]

Where the set includes all the constraints that are nonbinding and for which  at .

The optimal move size, , can now be determined by solving

[4.16a]

Subject to

[4.16b]

With any of the interval reduction methods of Section 4.1.

Figure 4.2.3 demonstrates the various type of constraints involved in this situation.

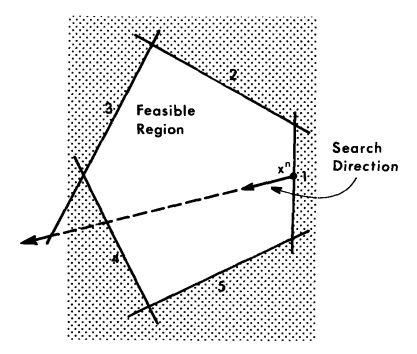


Figure 4.2.3 Maintaining feasibility along a search direction by bracketing range of the line search

**Finding a feasible descent direction.** In the cases where the current solution is on the boundary of the feasible region, the best descent direction may be pointing outside that region. In such case the direction has to be modified and the best feasible descent direction should be used to continue the search.

Those nonbinding constraints would not be violated by a small (infinitesimal) move in any direction and should not be considered. For the binding constraints, , where , the following must hold in order to ensure feasibility:

[4.17a]

Where is a small move size.

Since  for all , expression [4.17a] reduces to

[4.17b]

Condition [4.17b] should be satisfied by any feasible direction.

The direction offering the steepest feasible descent direction can be found by solving the program

 [4.18a]

Subject to

[4.18b][4.18c]

The objective function [4.18a] is the cosine of the angle between the gradient and the descent direction. This program then finds the descent direction that is closest to the opposite gradient direction yet satisfies the feasibility requirements. The last constraint normalizes the descent direction vector in order to ensure a unique solution. However, [4.18c] is a nonlinear constraint, meaning that the program is difficult to solve. The solution of this program is the steepest feasible descent direction.

**4.3 Convex combinations method**

The convex combination is known also as the Frank-Wolfe (FW) method, which is specially useful for determining the equilibrium flows for transportation networks.

**Algorithm**

The convex combinations algorithm is a ***feasible direction method***. However, the bounding of the move size does not require a separate step, which is accomplished as ***an integral part*** of the choice of the descent direction.

The direction-finding step of the convex combinations algorithm is explained in this section from the following two angles:

(1) this step is explained by using the logic of the general procedure for ***feasible direction methods*** outlined in the preceding section.

(2) this step is presented as a ***linear approximation method***.

Consider the convex program

[4.19a]

Subject to [4.19b]

Assume that at the iteration, the current solution is .

The convex combinations method selects the (feasible) descent direction not only on the basis of ***how steep each candidate direction is*** in the vicinity of , but also according to ***how far it is possible to move*** along this direction. The ***criterion for choosing direction*** in the convex combinations method is therefore based on the product of the rate of the descent in the vicinity of in a given direction and the length of the feasible region in that direction, which is known as the "drop". The algorithm uses the direction that ***maximizes the drop***.

To find descent direction, the algorithm looks at the entire feasible region for an auxiliary feasible solution, ,such that the direction from to provides the maximum drop. The direction from to any feasible solution, , is the vector  (or the unit vector ). The slope of the in the direction of is given by the projection this direction on the opposite gradient, that is,



The drop in the direction is obtained by multiplying this slope by the distance from to , , that is



This expression has to be maximized (in ) subject to the feasibility of . Alternatively, the expression can be multiplied by (-1) and minimized. Resulting in the program

[4.20a]

Subject to [4.20b]

Constraint [4.20b] represents that the point should be within the feasible region. Program [4.20] is a linear program.

The solution of the program [4.20] is , and the descent direction is the vector point from to , that is, , that is .

As mentioned in the beginning of this section, the another method is a linear approximation method. It is based on finding the descent direction by minimizing a linear approximation to the function (instead of the function itself ) at the current solution point. Let denotes the linear approximation of the value of the objective function at some point , based on its value at . The approximation is given by (Taylor formula at the first order derivative of )



This linear function of has to be minimized subject to the constraints of the original problem, that is

[4.22a]

Subject to

[4.22b]

Note that the value of the objective function at point , is a constant, which can be dropped from Eq.[4.22a]. Thus this program is identical to program [4.20].

The objective function of the linearized program, can, however, be simplified even further by noting that is a constant at and the term can therefore be dropped from the linearized program, which can be written as

[4.33a]

Subject to

[4.33b]

The solution of program [4.23] is used to define the descent direction (i.e.).

As mentioned before, the convex combinations methods does not require a special step to bracket the search for an optimal move size to maintain feasibility. The new solution, , must lie between and (since  being a solution of a linear program, naturally lies at the boundary of the feasible region). Since the search interval is bracketed, then, any of the interval reduction methods would be suitable for the minimization of along , that is, for solving



Subject to



Once the optimal solution of this line search, , is found, the next point can be generated with the usual step,

[4.25]

Note that the Eq. [4.25] can be written as . The new solution is thus a convex combination of and .

The convergence criterion can be base on the similarity of two successive solutions or the reduction of the objective function values between successive iterations.

Given a current feasible solution, , the iteration of the convex combinations algorithm can be summarized as follows:  
Step 1 *Direction finding.* Find  that solves the linear program [4.23]

Step 2 *Step-size determination.* Find  that solves



Step 3 *move*. Set 

Step 4 *convergence test*. If , stop. Otherwise, let  and go to step 1.