On the choice of step size in subgradient optimization

Mokhtar S. BAZARAA
School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A.

Hanif D. SHERALI
School of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, U.S.A.

Received October 1979
Revised January 1980

This paper recommends some procedures for the selection of step sizes in the context of subgradient optimization. The first of these procedures is developed in detail in this study and is a theoretically convergent scheme. This method has two phases, the first phase is designed to accelerate the solution procedure towards an optimal solution, while the second phase helps to close in on an optimal solution. A second technique recommended is a simple-minded scheme which, although not theoretically convergent, seems to be computationally very efficient. These two methods are shown to compare favorably with Held, Wolfe and Crowder's scheme for prescribing step sizes. We also suggest some modifications of the latter scheme to make it computationally more efficient.

1. Introduction

Subgradient optimization is one of the most widely used tools in the field of mathematical programming, especially within the contexts of Lagrangian relaxation and branch and bound. Despite its acceptability among researchers and despite the availability of ample theoretical convergence results, many researchers have experienced erratic behavior of the method as different rules for choosing the step size along the subgradient vector are used.

The purpose of this paper is to recommend specific rules for selecting step sizes which appear to be computationally desirable. Towards this end, we first develop a two phase scheme which blends the step size strategies of Poljak [9,10]. This procedure is designed so that the first phase accelerates the method towards an optimal solution, and the second phase not only attains theoretical convergence, but also helps the procedure computationally in rapidly closing in on the optimal objective value. A second method recommended is a simplistic scheme which appears to be computationally very efficient. We also test Held, Wolfe and Crowder's [5] technique for selecting step sizes and recommend how one may modify this suitably so as to obtain a more rapid rate of initial improvement in the objective function value.

North-Holland Publishing Company
European Journal of Operational Research 7 (1981) 380–388

2. Fundamental concepts in subgradient optimization

Consider the following problem D, where $\theta : R^n \to R^1$ is concave and W is a closed convex set in R^n.

$$\text{D}: \quad \text{maximize} \quad \theta(w)\,,$$
$$\text{subject to} \quad w \in W$$

A typical subgradient optimization scheme for solving the above problem may be stated as follows. Starting with an arbitrary solution in W, at any stage k, given $w_k \in W$, let $\partial\theta(w_k)$ denote the set of subgradients of θ at w_k. Let $\xi_k \in \partial\theta(w_k)$, and choose a suitable step size $\lambda_k > 0$. Let $w_{k+1} = P_W(w_k + \lambda_k \xi_k)$, replace k by $k+1$, and repeat the process. Here, $P_W(\cdot)$ denotes the projection operator on W and is defined by

$$P_W(z) = y \leftrightarrow y \in W \text{ and } \|y - z\| \leqslant \|w - z\| \text{ for all } w \in W\,.$$

The basis of the above scheme lies in the well known fact that given a non-optimal solution $w_k \in W$ and any $\xi_k \in \partial\theta(w_k)$, there exists a step size $\lambda_k > 0$ such that $P_W(w_k + \lambda_k \xi_k)$ is closer, in the Euclidean norm sense, to an optimal solution to Problem D than w_k is. However, in order to obtain convergence to an optimal solution, it is not necessary to find a step size λ_k which performs ideally with respect to the above criterion. In fact, Poljak [9] has shown that the conditions

$$\lambda_k \|\xi_k\| \to 0 \quad \text{and} \quad \sum_k \lambda_k \|\xi_k\| = \infty \tag{2.1}$$

0377-2217/81/0000–0000/$02.50 © North-Holland

are sufficient to guarantee that lim sup $\theta(w_k) = \sup_{w \in W} \theta(w)$.

From the point of view of implementation of the subgradient scheme, two requirements are desirable. First, an easy method of computing a subgradient vector $\xi_k \in \partial\theta(w_k)$ must be available, and second, W must be simple enough to admit an easy projection. Particularly, if we consider the problem of minimizing $f(x)$ subject to $g_i(x) \leqslant 0$ for $i = 1, ..., n$ and $x \in X$, then the Lagrangian dual problem is given by Problem D above with

$$\theta(w) = \inf\{f(x) + w^t g(x): x \in X\}, \qquad W = \{w: w \geqslant 0\},$$

where a superscript t denotes the transpose operation. In this case, given $w \geqslant 0$, if $\hat{x} \in X$ satisfies $\theta(w) = f(\hat{x}) + w^t g(\hat{x})$, then $\xi = g(\hat{x})$ is a subgradient vector of θ at w. Furthermore, the projection $P_W(z)$ is given by the vector y whose components are defined by $y_i = \max\{0, z_i\}$ for $i = 1, ..., n$. It is perhaps due to this ease of computing subgradients and projections that subgradient optimization is so widely used within the context of Lagrangian relaxation.

Now, although the conditions given in (2.1) are sufficient to guarantee convergence, not all choices of step sizes $\lambda_k > 0$ satisfying (2.1) may be desirable. The reason being that a very slow convergence rate may result. For example, in a study conducted by Edwards [3], the choice $\lambda_k \|\xi_k\| = 1/k$ for $k = 1, 2, ...$ performs very poorly. In order to overcome this difficulty, Poljak [9] devised the following alternative step size rule. At each stage k, select

$$\lambda_k = \delta_k \left(\frac{\theta^* - \theta(w_k)}{\|\xi_k\|^2} \right), \qquad \epsilon_1 < \delta_k < 2 - \epsilon_2, \tag{2.2}$$

where $\theta^* = \sup\{\theta(w): w \in W\}$ and $\epsilon_1, \epsilon_2 > 0$ are suitably chosen to admit a non-empty range of positive values for δ_k. Poljak [10] has shown that the step rule (2.2) guarantees convergence, and under certain other assumptions, the convergence is realized at a geometric rate. Note, however, that in order to apply the rule specified in (2.2) one has to know θ^* a priori, which is impossible for most problems. Poljak suggested that θ^* be replaced by $\bar{\theta} < \theta^*$. He proved, in this case, that the sequence generated is such that $\theta(w_k) > \bar{\theta}$ for some k or else $\theta(w_k) \leqslant \bar{\theta}$ for all k and $\theta(w_k) \to \bar{\theta}$. In either case, however, it is clear that we have no assurance of converging to θ^*.

We will now proceed to develop a method for suitably selecting $\lambda_k > 0$ satisfying (2.1) which tends to accelerate the approach of w_k towards an optimal solution w^* to Problem D. It may be worthwhile noting here that according to the computational experience of Held, Wolfe and Crowder [5], the second condition in (2.1) does not seem to be overly important in obtaining numerical convergence. However, the scheme proposed in the following section is not designed merely to satisfy (2.1), but as computational results indicate, each phase of the procedure helps in rapidly obtaining convergence.

3. Selection of step sizes to accelerate convergence

In this section, we will concentrate on selecting a suitable step size $\lambda_k > 0$ along the subgradient vector $\xi_k \in \partial\theta(w_k)$, where $w_k \in W$ is given, in an attempt to satisfying the criterion of obtaining $w_k + \lambda_k \xi_k$ as close as possible to an optimal solution w^* to Problem D, by moving along ξ_k from w_k. The ideal value of λ_k in this respect is clearly an optimal solution to the problem

$$\text{minimize} \quad \|w^* - (w_k + \lambda\xi_k)\|^2 .$$

It may be easily verified that this optimal solution is given by

$$\lambda_k^* = \frac{\xi_k^t(w^* - w_k)}{\|\xi_k\|^2} \tag{3.1}$$

Now, by the concavity of θ we have

$$\lambda_k^* = \frac{\xi_k^t(w^* - w_k)}{\|\xi_k\|^2} \geqslant \frac{\theta^* - \theta(w_k)}{\|\xi_k\|^2}. \tag{3.2}$$

From (3.2), a reasonable approach to approximate λ_k^* is to use a step size

$$\lambda_k = \frac{\bar{\theta} - \theta(w_k)}{\|\xi_k\|^2}, \tag{3.3}$$

where $\bar{\theta} \geqslant \theta^*$, which conforms with the suggestion of Held, Wolfe and Crowder [5]. Rather than choosing a fixed $\bar{\theta} \geqslant \theta^*$, however, we would like to update $\bar{\theta}$ periodically such that $(\bar{\theta} - \theta^*)/\|\xi_k\|^2$ is directly related to the gap in the inequality of (3.2). Since θ is concave, this gap is more pronounced when w_k is farther away from the set of optimal solutions to the problem. Thus we propose the step size given by (3.3) with $\bar{\theta}$ given by:

$$\bar{\theta} = \alpha_r\theta^0 + (1 - \alpha_r)\theta^c, \tag{3.4}$$

where θ^0 is a fixed number such that $\theta^0 > \theta^*$, θ^c is the current best objective value, and where the monotone decreasing sequence $\{\alpha_r\}$ has the property that:

$$\alpha_0 = 1, \qquad \alpha_r \to \epsilon_0 > 0 . \tag{3.5}$$

Note that we would like $\{\alpha_r\}$ to maintain $\bar{\theta}$ as an overestimate of θ^*. Although we are not able to guarantee this, we do have the desirable property that $\bar{\theta}$ is bootstrapped upwards by an improved value of θ^c. Hence, we may relate the monotone decreasing sequence $\{\alpha_r\}$ to the sequence $\{w_k\}$ in the following manner. If for several iterations, the value of θ^c does not improve, then it is likely that we are taking step sizes which are too large, and hence we need to reduce $\bar{\theta}$ as seen from (3.3). Thus, we increment r by one in (3.4) causing α_r and $\bar{\theta}$ to decrease. Further, note that we have stipulated a positive limit ϵ_0 for the sequence $\{\alpha_r\}$ in (3.5). As we shall see later, this serves as an expedient in not only obtaining theoretical convergence, but also in our computational experience, in increasing the rapidity of convergence.

It is apparent that if we desire to avoid the frequent occurrence of $\bar{\theta}$ turning out to be an underestimate of θ^*, we must select a sequence α_r which weights θ^0 heavily in the initial stages of the process when θ^c is likely to be significantly less than θ^* and which shifts the weight onto θ^c gradually at first and then more rapidly later as θ^c approaches θ^*. We may hence be interested in a function $\alpha : R \to R$ whose graph is shown in Fig. 1. An analytical representation of such a relationship is:

$$\alpha(x) = e^{-px^q} \tag{3.6}$$

where p and q are suitable parameters

The choice of the parameters p and q may be governed by the following considerations. Let us first differentiate $\alpha(\cdot)$ twice to obtain

$$\alpha'(x) = -pqx^{q-1}e^{-px^q}, \tag{3.7}$$

$$\alpha''(x) = pqx^{q-2}e^{-px^q}[pqx^q - (q-1)]. \tag{3.8}$$

Now observe from (3.7) and (3.8) that if $p > 0, q > 1$, $x \geqslant 0$, the rate of decrease of α starts from zero and gradually reaches a maximum at the threshold point given by

$$\hat{x} = \left[\frac{(q-1)}{pq}\right]^{1/q} \tag{3.9}$$

and then rapidly begins to taper off to zero. This is precisely the property we were seeking. At the threshold point $\hat{x} = r_1$, we obtain

$$\alpha(r_1) = e^{-(q-1)/q}. \tag{3.10}$$

Let us arbitrarily agree to let r_1 be integral and stipulate $\alpha(r_1) = \frac{1}{2}$, that is, let us allow $\alpha(x)$ to decrease gradually from 1 to $\frac{1}{2}$ as x increases from zero to r_1, and then decrease more rapidly thereafter. Observe that we still have control over the procedure through the parameter r_1, by which we may delay, or hasten, the shift of the dominating weight from θ^0 to θ^c. Hence, we have:

$$\alpha(r_1) = e^{-(q-1)/q} = \tfrac{1}{2} \quad \text{yielding } q = 3.26. \tag{3.11}$$

Substituting this value of q into (3.9), where $\hat{x} = r_1$, we have

$$p = 0.6933/r_1^{3.26}. \tag{3.12}$$

Thus, from (3.6), (3.11), (3.12), we have

$$\alpha(x) = e^{-0.6933(x/r_1)^{3.26}}. \tag{3.13}$$

Now, let us select a suitable value of ϵ_0 as the minimum weight α_r specified in (3.5). Further, let us determine r_2 as the smallest integer which satisfies

$$\epsilon_0 \geqslant e^{-0.6933(r_2/r_1)^{3.26}}. \tag{3.14}$$

Then, our choice of the sequence $\{\alpha_r\}$ in view of (3.5) is thus given by

$$\alpha_r = \begin{cases} e^{-0.6933(r/r_1)^{3.26}}, & \text{if } r < r_2, \\ \epsilon_0, & \text{if } r \geqslant r_2. \end{cases} \tag{3.15}$$

In Section 4, we recommend suitable values for the parameters r_1 and ϵ_0. Typically, if θ^0 is close to θ^*, a relatively higher value for r_1 and a relatively lower value for ϵ_0 are desirable and vice-versa otherwise.

We are now ready to discuss our procedure which consists of two phases. Phase I is an acceleration phase and the step sizes are determined according to (3.3), where $\bar{\theta}$ is periodically updated, and Phase II, assures that the convergence conditions of Poljak given by (2.1) are satisfied without slowing down the convergence rate. We begin the procedure by setting $r = 0$, choosing a suitable positive integer r_1 for (3.15), and

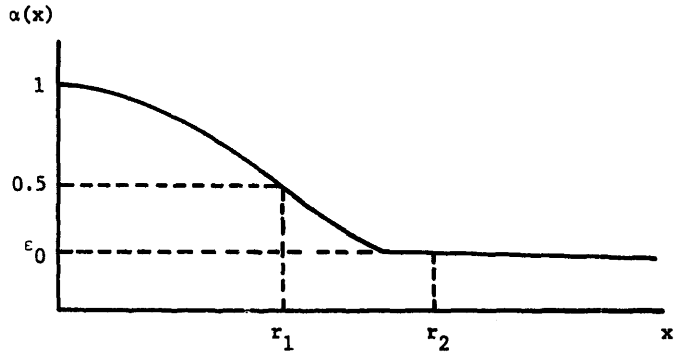

Fig. 1. Graph of the α function.

determining an arbitrary solution $w_1 \in W$. In general, given a solution w_k, we determined a subgradient ξ_k of θ at w_k. We then compute the step size λ_k from (3.3), (3.4), and hence, arrive at the solution $w_{k+1} = P_W(w_k + \lambda_k \xi_k)$. For the sake of convergence, we will call this iteration a success or failure according to whether or not $\theta(w_{k+1}) - \theta^c \geq \epsilon > 0$, where ϵ is a prespecified fixed tolerance level. If we have experienced $\bar{\nu}$ consecutive failures, where $\bar{\nu}$ is a pre-specified integer parameter, we increment r by one, update α_r using (3.15) and reset to the current best solution. This particular resetting prevents the sequence $\{w_k\}$ from straying over too many iterations without improving. In any event, if $r < r_2$, we repeat the above step of Phase I. On the other hand, when r becomes equal to r_2, we stop incrementing r and we freeze α_r at ϵ_0 hereafter. The procedure passes onto Phase II at this point.

In Phase II of this procedure, we have

$$\bar{\theta} = \epsilon_0 \theta^0 + (1 - \epsilon_0)\theta^c \tag{3.16}$$

and we select step sizes according to

$$\lambda_k = \frac{1}{\beta_k}\left[\frac{\bar{\theta} - \theta(w_k)}{\|\xi_k\|^2}\right], \tag{3.17}$$

where $\beta_k = 1$ initially, and is incremented by two every $\bar{\nu}$ iterations, regardless of a success or a failure. Note that resetting in Phase II is performed whenever β_k is incremented so long as β_k is less than some fixed integer $\bar{\beta} \geq 1$. Thereafter, no resetting to the current best solution is performed. We now formalize the above statements in the format of an algorithmic scheme:

Initialization. Choose a starting solution $w_1 \in W$ and compute $\theta(w_1)$. Determine $\xi_1 \in \partial\theta(w_1)$. If $\|\xi_1\| = 0$, stop with w_1 as an optimal solution to the problem. Otherwise, let $\theta^0 > \theta^* = \sup\{\theta(w): w \in W\}$ and set $\bar{\theta} = \theta^0$. Let $(w^c, \xi^c, \theta^c) = (w_1, \xi_1, \theta(w_1))$. Select appropriate positive values for ϵ_0, and ϵ, and select appropriate positive integral values for $\bar{\nu}, \bar{\beta}$, and r_1. Let $\beta_1 = 1$, $r = 0$, $\nu = 0$, $k = 1$, $\alpha_0 = 1$ and go to Step 1.

Step 1. Given w_k, β_k, $\theta(w_k)$, ξ_k and $\bar{\theta}$, determine

$$\lambda_k = \frac{1}{\beta_k}\left[\frac{\bar{\theta} - \theta(w_k)}{\|\xi_k\|^2}\right].$$

Let $w_{k+1} = P_W(w_k + \lambda_k \xi_k)$. Compute $\theta(w_{k+1})$ and determine $\xi_{k+1} \in \partial\theta(w_{k+1})$. If $\|\xi_{k+1}\| = 0$, terminate with w_{k+1} as an optimal solution to the problem. Otherwise, replace k by $k+1$ and go to Step 2 if $r < r_2$ or to Step 5 if $r = r_2$.

Phase I

Step 2. If $\theta(w_k) \geq \theta^c + \epsilon$, go to Step 4; otherwise, go to Step 3.

Step 3. Replace ν by $\nu + 1$. If $\nu < \bar{\nu}$, go to Step 1. If $\nu = \bar{\nu}$, set $\nu = 0$ and replace r by $r + 1$. Compute

$$\alpha_r = e^{-0.6933(r/r_1)^{3.26}}$$

and set

$$\bar{\theta} = \alpha_r \theta^0 + (1 - \alpha_r)\theta^c .$$

Let $\beta_k = \beta_{k-1}$, reset $(w_k, \xi_k, \theta(w_k))$ to (w^c, ξ^c, θ^c) and go to Step 1.

Step 4. Let $(w^c, \xi^c, \theta^c) = (w_k, \xi_k, \theta(w_k))$, and compute $\bar{\theta} = \alpha_r \theta^0 + (1 - \alpha_r)\theta^c$. Set $\nu = 0$, $\beta_k = \beta_{k-1}$, and go to Step 1.

Phase II

Step 5. If $\theta(w_k) > \theta^c$, go to Step 6. Otherwise, go to Step 7.

Step 6. Replace (w^c, ξ^c, θ^c) by $(w_k, \xi_k, \theta(w_k))$. Compute $\bar{\theta} = \alpha_r \theta^0 + (1 - \alpha_r)\theta^c$ and go to Step 7.

Step 7. Replace ν by $\nu + 1$. If $\nu < \bar{\nu}$, go to Step 1. If $\nu = \bar{\nu}$, set $\nu = 0$, and let $\beta_k = \beta_{k-1} + 2$. Now, if $\beta_k \geq \bar{\beta}$ return to Step 1. Otherwise, reset $(w_k, \xi_k, \theta(w_k))$ to (w^c, ξ^c, θ^c) and return to Step 1.

The following theorem shows that the above procedure converges to the optimal objective value of the problem.

Theorem. *Consider the problem to maximize $\theta(w)$ subject to $w \in W$, where θ is a concave function and W is a closed convex set. Suppose that $\theta^* = \sup\{\theta(w): w \in W\}$ is finite and that the sequence $\{w_k\}$ generated by the algorithm is contained in a compact set $\hat{W} \subset W$. Then the sequence $\{\theta^c\}$ of the current best objective values converges to θ^*.*

Proof. First consider the case that there exists a set of positive integers indexed by $\mathcal{K}$ so that $\xi_k \to 0$. By compactness of $\hat{W}$, there exists a point $\hat{w} \in \hat{W}$ and an index set $\mathcal{K}' \subset \mathcal{K}$ so that $w_k \to \hat{w}$. Therefore, $\xi_k \to 0$ so that 0 is a subgradient of θ at $\hat{w}$. This, in turn, implies that $\theta(\hat{w}) = \theta^*$ and since $\theta(w_k) \to \theta(\hat{w}) = \theta^*$ it is clear that $\theta^c \to \theta^*$. Now consider the case when $\{\xi_k\}$ is bounded away from 0, that is, suppose that there exists a number $z > 0$ so that

$$\|\xi_k\| \geq z > 0 \qquad \text{for all } k . \tag{3.18}$$

We first show that there exists an integer K such that $r = r_2$ for all $k \geq K$. If not, then after r reaches its maximum which is strictly less than r_2, θ must increase by at least $\epsilon > 0$ every $\bar{\nu}$ iterations. This contradicts our assumption that θ^* is finite. Thus, for $k \geq K$ the procedure is in Phase II with $\alpha_r = \epsilon_0$ and the step size λ_k given by:

$$\lambda_k = \frac{1}{\beta_k}\left[\frac{\epsilon_0(\theta^0 - \theta^c) + (\theta^c - \theta(w_k))}{\|\xi_k\|^2}\right]. \tag{3.19}$$

To establish the fact that $\theta^c \to \theta^*$, in view of Poljak's theorem [9], it suffices to show that $\lambda_k\|\xi_k\| \to 0$ and that $\Sigma_k \lambda_k \|\xi_k\| \to \infty$. Noting compactness of $\hat{W}$, it is clear that the numerator in (3.19) is bounded above. Thus, from (3.18) and the fact that β_k is increased by two every $\bar{\nu}$ iterations, it is clear that $\lambda_k\|\xi_k\| \to 0$ as $k \to \infty$. Now, we will show that $\Sigma_k \lambda_k\|\xi_k\| \to \infty$. Since $\{w_k\}$ is contained in a compact set $\hat{W}$, there exists a number $\delta > 0$ so that $\|\xi_k\| \leq \delta$ for each k, see Rockafellar [11, p. 237]. Therefore (3.19) yields

$$\sum_k \lambda_k\|\xi_k\| \geq \sum_k \frac{1}{\beta_k \delta}[\epsilon_0(\theta^0 - \theta^c) + (\theta^c - \theta(w_k))] .$$

But $\theta^c - \theta(w_k) \geq 0$ and $\theta^0 - \theta^c \geq \theta^0 - \theta^*$. Thus,

$$\sum_k \lambda_k\|\xi_k\| \geq \epsilon_0\left[\frac{\theta^0 - \theta^*}{\delta}\right]\sum_k \frac{1}{\beta_k} .$$

Since $\epsilon_0(\theta^0 - \theta^*/\delta)$ is a fixed positive constant and $\Sigma_k 1/\beta_k$ diverges, the proof is complete.

4. Computational testing and further recommendations

In this section, we will conduct a numerical investigation of three step size strategies using test problems available in the literature. As reported by Edwards [3], subgradient optimization has generally proved inadequate for continuous nonlinear programming problems primarily due to the difficulty that the subproblem used for evaluating $\theta(w_k)$ and ξ_k is usually nonconvex. This usually causes stopping at a local optimal solution which results in an inaccurate evaluation of $\theta(w_k)$ and ξ_k. In view of this, discrete and linear programming problems are used for testing the step size rules. Specifically, we use the Travelling Salesman Problem and the Linear Assignment Problem to test the procedures using data from the published literature [1,2,4,8]. The first strategy is the one proposed in the foregoing section. The second strategy is the one proposed by Held, Wolfe and Crowder [5] with one minor modification. Whereas Held, Wolfe and Crowder recommend that

$$\lambda_k = \delta_k\left[\frac{\theta^0 - \theta(w_k)}{\|\xi_k\|^2}\right], \tag{4.1}$$

where $\delta_1 = 2$ initially for $2m$ iterations, with m designating the number of facilities or cities in the context of our test problems, we will maintain $\delta_1 = 2$ for m iterations to start with. The rest of the procedure remains the same. In our experience, this modification improves the rate of improvement of the objective function. In fact, after presenting the computational results, we will recommend a further change in this scheme. The third strategy commences with an initial step size equal to that given through the first strategy. Thereafter, everytime the procedure goes through $\bar{\nu}$ consecutive failures, the step size is halved and the procedure is reset to the current best solution. Otherwise, the step size is maintained unchanged. Henceforth, these step size strategies will be respectively referred to as strategies 1, 2 and 3.

Three termination criteria are adopted for computational purposes. The procedures are terminated if any one of the following occur:

(1) $\|\xi_k\| = 0$ at any iteration k.
(2) The iteration index k equals 200. (4.2)
(3) $\lambda_k\|\xi_k\| \leq 10^{-5}\sqrt{m}$ for 4 consecutive iterations.

Further, the following parameters are chosen for all schemes. The starting vector $w_1 \in W$ is chosen as the zero vector, $\bar{\nu}$ is taken equal to three, and ϵ is taken as 0.001. For the step size strategy 1, $\bar{\beta}$ is taken as 120 throughout and for step size strategy 2, δ_k of (4.1) is halved every 6 iterations [5].

4.1. The symmetric travelling salesman problem

The symmetric travelling salesman problem can be stated as follows, where c_{ij} is the cost of link (i,j):

$$\text{minimize} \quad \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}x_{ij} ,$$

$$\text{subject to} \quad \sum_{j=l}^{m} x_{ij} = 1 , \quad i = 1, ..., m , \tag{4.3}$$

$$\sum_{i=1}^{m} x_{ij} = 1 , \quad j = 1, ..., m , \tag{4.4}$$

$$x_{ij} = 0 \text{ or } 1 , \quad i,j = 1, ..., m , \tag{4.5}$$

No subtours.

Letting X be the set of all 1-trees (see [6,7]), subtours can be eliminated by insisting that a vector x satisfying the constraints (4.3), (4.4) and (4.5) must also belong to X. In particular for the symmetric case, constraints (4.3) and (4.4) can be replaced by (4.6) leading to the following equivalent formulation of the symmetric travelling salesman problem.

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} \sum_{\substack{j=1 \\ j\neq i}}^{m} c_{ij}x_{ij} , \\ &\text{subject to} \quad \sum_{\substack{j=1 \\ j\neq i}}^{m} x_{ij} + \sum_{\substack{j=1 \\ j\neq i}}^{m} x_{ji} = 2 \qquad \text{for } i = 1, ..., m, \\ &\qquad\qquad x \in X . \end{aligned} \tag{4.6}$$

Incorporating the restrictions given by (4.6) into the objective function via the Lagrangian multiplier vector w, we obtain the following problem D with $W = R^{m}$:

maximize $\theta(w)$,

where

$$\theta(w) = -2 \sum_{i=1}^{m} w_i + \min\{\sum_{i=1}^{m} \sum_{\substack{j=1 \\ j\neq i}}^{m} (c_{ij} + w_i + w_j)x_{ij} : x \in X\} .$$

Furthermore, it may be easily verified that given a vector $\overline{w}$, if $\overline{x} \in X$ determines $\theta(\overline{w})$, then ξ whose ith component is given by

$$\xi_i = \sum_{\substack{j=1 \\ j\neq i}}^{m} \overline{x}_{ij} + \sum_{\substack{j=1 \\ j\neq i}}^{m} \overline{x}_{ji} - 2 , \quad i = 1, ..., m$$

is a subgradient of θ at $\overline{w}$. A greedy, one-pass heuristic was used to determine a tour, the length of which was taken as θ^0. Finally, for step size strategy 1, we found it suitable to use $r_1 = 3$ and $\epsilon_0 = 0.1$. Table 1 gives the computational experience on a CDC Cyber 74 computer with coding in FORTRAN IV. The data for the 10-city problem is from Barachet [1], that for the 25 and 48 city problems are from Held and Karp [4], the 33 city and the 57 city problem data are from Karg and Thompson [8], and the 42 city problem is from Dantzig, Fulkerson and Johnson [2].

The results show that step size strategies 1 and 3 are fairly rapid in improving the objective function values in the initial stages. This is particularly important in the context of branch-and-bound where a procedure that significantly increases the objective function in a few number of iterations is usually preferred to one that yields a high objective value, but only after a large number of iterations are performed. From Table 1 it is apparent that strategies 1 and 3 give a large bound after 25–40 iterations have been expended. The initial rate of improvement of strategy 2 is very slow, however, principally because the procedure is overstepping. Even when the overestimate θ^0 is obtained with a minimal amount of effort, this technique may be improved by starting with δ_k held at a smaller value, say 1, at a fewer number of iterations, say $\frac{1}{2}m$. This remark is apparent from the results in Table 1. On the whole, although strategy 1 leads to a theoretically convergent scheme and performs fairly well computationally, strategy 2 can be modified as above to operate comparably, but strategy 3 appears to work best and is moreover very straightforward.

4.2. The linear assignment problem

The linear assignment problem can be stated as follows:

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij}x_{ij} , \\ &\text{subject to} \quad \sum_{j=1}^{m} x_{ij} = 1 , \qquad i = 1, ..., m , \\ &\qquad\qquad \sum_{i=1}^{m} x_{ij} = 1 , \qquad j = 1, ..., m , \\ &\qquad\qquad x_{ij} \geqslant 0 , \qquad i,j = 1, ..., m . \end{aligned} \tag{4.7}$$

Incorporating the constraints given by (4.7) into the objective function through the use of w_i for $i = 1, ...$, m, the dual problem can be stated as follows, with $W = R^{m}$:

maximize $\theta(w)$,

where

$$\begin{aligned} \theta(w) = \text{minimum} \; &\sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{i=1}^{m} w_i(\sum_{j=1}^{m} x_{ij} - 1) , \\ &\text{subject to} \sum_{i=1}^{m} x_{ij} = 1 , \qquad j = 1, ..., m , \\ &\qquad x_{ij} \geqslant 0 , \qquad i,j = 1, ..., m . \end{aligned} \tag{4.8}$$

Table 1
The travelling salesman problem

Step size strategy	m	known θ^*	$\theta(w_1)$	θ^0	a	b	c	d	e	f	Approximate values of $\theta(\cdot)$ at the 5	10	15	20
	10	378	304	411	40	378.00	40	25	376.6	1	361.5	361.5	364.9	373.6
	25	1711	1453	2037	50	171.1	50	30	1705.2	1	1453.0	1500.5	1594.0	1672.3
	33	10861	10007	11994	38	10861	38	32	10825.8	1	10235.3	10513.2	10513.2	10619.2
1	42	697	629	699	200	693.47	200	68	692.5	2	658.6	666.1	673.9	675.1
	48	11444.5	10439	14241	200	11442.6	185	22	11293.6	2	10439	10605	10983.3	11246
	57	12907.5	11520	15889	200	12905.78	168	36	12792	2	11520	11986.6	12070.2	12338.6
	10	378	304	411	68	378.0	62			1	309	333.1	360.6	371.7
	25	1711	1453	2037	77	1711	77			1	1453	1453	1453	1453
2	33	10861	10007	11994	86	10861	86			1	10007	10007	10007	10007
	42	697	629	699	200	692.1	199	N/A	N/A	2	629	629	629	629
	48	11444.5	10439	14241	200	11437.4	198			2	10439	10439	10439	10439
	57	12907.5	11520	15889	200	12906.5	196			2	11520	11520	11520	11520
	10	378	304	411	43	378.0	43			1	361.5	364.9	375.3	375.4
	25	1711	1453	2037	38	1711	38			1	1453	1632.4	1658.3	1688.8
3	33	10861	10007	11994	42	10861	42	N/A	N/A	1	10211.4	10300.5	10619.2	10761.1
	42	697	629	699	200	692.1	198			2	658.6	666.1	675.5	678.8
	48	11444.5	10439	14241	172	11437.8	151			3	10439	10985	11266.2	11319.5
	57	12907.5	11520	15889	197	12902.8	179			3	11520	12112.3	12524.2	12677.1

a – Total number of iterations.
b – Best recorded solution value.
c – Iteration number at which the best recorded solution was detected.
d – Iteration number at which the procedure switches from Phase I to Phase II for step size strategy 1.
e – Value of $\theta(\cdot)$ at the switchover in d.
f – Termination type number defined in Eq. (4.2).

Table 2
The linear assignment problem

Step size strategy	m	known θ^*	$\theta(w_1)$	θ^0	a	b	c	d	e	f	Approximate values of $\theta(\cdot)$ at the 5	10	15	20
	10	326	251	362	35	326	35	–	–	1	309.8	309.8	311.6	321.4
	25	1281	1061	1470	200	1280.97	152	46	1277.9	2	1158.7	1158.7	1158.7	1191.2
1	33	9948	8590	11219	200	9947.9	194	44	9903.7	2	8590	8590	8742	9251
	42	532	454	581	200	531.99	163	62	530.9	2	487	487	487	496.4
	48	9870	8757	14072	200	9869.2	198	47	9771.9	2	8818.8	8827.5	8827.5	8827.5
	57	10553	8928	12311	200	10552.6	189	51	10491.4	2	9263.8	9263.8	9263.8	9519.2
	10	326	251	362	27	326	27			1	297.9	297.9	321.4	322.9
	25	1281	1016	1470	183	1280.99	133			3	1016	1016	1016	1016
2	33	9948	8590	11219	200	9947.99	188	N/A	N/A	2	8590	8590	8590	8590
	42	532	454	581	198	531.99	149			3	454	454	454	454
	48	9870	8757	14072	200	9860.0	195			2	8757	8757	8757	8757
	57	10553	8928	12311	200	10528.4	196			2	8928	8928	8928	8928
	10	326	251	362	19	326	19			1	309.75	323.0	324.5	–
	25	1281	1016	1470	134	1280.99	116			3	1158.7	1233	1255.8	1260.9
3	33	9948	8590	11219	147	9947.99	123	N/A	N/A	3	8999.2	9548.7	9657.4	9848.7
	42	532	454	581	120	531.99	99			3	486.98	508.7	517.6	523.8
	48	9870	8757	14072	162	9869.98	141			3	8818.8	8972.4	9355.6	9604.7
	57	10553	8928	12311	167	10534.57	143			3	9263.8	9549.9	10054.96	10221.8

a, b, c, d, e and f have the same connotation as for Table 1 llowing iterations:

25	30	40	50	75	100	125	150	175	cpu seconds
376.6	377.5	378.0	–	–	–	–	–	–	0.209
695.4	1705.2	1710.9	1711.0	–	–	–	–	–	1.207
728.0	10819.8	–	–	–	–	–	–	–	1.848
681.3	682.9	686.9	691.0	692.7	693.1	693.3	693.4	693.4	9.765
337.2	11363.4	11414.6	11420	11427.8	11434.5	11437.9	11439.6	11442.0	12.686
650.9	12689.4	12837.8	12861.3	12888.7	12896.4	12900.2	12904.3	12905.8	18.116
374.6	375.4	377.2	377.7	–	–	–	–	–	0.242
453	1453	1543.2	1688.0	1710.9	–	–	–	–	1.574
007	10007	10235	10545.8	10845.6	–	–	–	–	2.961
629	629	629	629	629	673.9	673.9	688.5	691.9	9.446
439	10439	10439	10439	10646.7	11358.1	11426.9	11436.6	11437.4	12.522
520	11520	11520	11520	11520	12400.7	12846.3	12902.8	12906.2	18.156
76.9	377.3	377.9	–	–	–	–	–	–	0.16
00.7	1703.5	–	–	–	–	–	–	–	0.973
69.9	10846.5	10860.1	–	–	–	–	–	–	1.768
81.7	684.0	686.1	688.0	690	691	691.5	691.8	692	9.619
78.1	11409.7	11416.3	11425.8	11436.2	11437.5	11437.7	11437.8	–	10.908
32.9	12787.9	12850.3	12883.4	12899.6	12901.8	12902.6	12902.7	12902.7	17.312

llowing iterations:

25	30	40	50	75	100	125	150	175	cpu seconds
322.0	323.5	–	–	–	–	–	–	–	0.152
240	1241.3	1271.8	1279.9	1280.8	1280.9	1280.96	1280.96	1280.96	3.978
580.3	9585.9	9903.7	9431.3	9946	9947	9947.5	9947.8	9947.8	7.633
498.6	504.8	520.2	528.3	531.6	531.9	531.96	531.98	531.98	12.006
005.8	9378.6	9731.7	9844.3	9860.3	9864.6	9867.8	9868.4	9869.2	15.246
044.9	10108.2	10423.8	10491.4	10533.6	10543.4	10548.2	10550.7	10552.1	21.398
322.9	–	–	–	–	–	–	–	–	0.163
016	1120.5	1194.2	1257.5	1277.9	1280.8	1280.9	–	–	4.24
590	8590	8590	8590	9889.1	9945.2	9947.8	9947.96	9947.99	7.044
454	454	454	486.9	517.6	531.3	531.9	531.99	531.99	12.101
757	8757	8757	8818.8	8818.8	9553.9	9853.6	9857.6	9859.9	14.387
928	8928	8928	8928	9263.8	10054.96	10497.9	10524.6	10527.99	20.142
–	–	–	–	–	–	–	–	–	0.080
263.2	1273.8	1277.6	1279.3	1280.8	1280.97	1280.97	–	–	3.070
848.7	9895.1	9921.4	9936.9	9947.2	9947.97	9947.97	–	–	5.611
528.2	529.2	530.7	531.58	531.96	631.99	–	–	–	7.141
630.7	9767.5	9830.3	9852.5	9867.87	9869.76	9869.95	9865.95	–	11.851
299.8	10302.9	10481.6	10513.2	10529.6	10533.8	10534.5	10534.56	–	16.223

In order to compute $\theta(w)$, it is clear that an optimal solution to the problem stated in (4.8) is given as follows. For any given j, let

$$I_j = \{i\colon c_{ij} + w_i = \min_{1\leq k\leq m} c_{kj} + w_k\} .$$

Then for each component j, an optimal solution $\overline{x}$ is obtained by setting $\overline{x}_{ij} = 1$ for one index $i \in I_j$ while setting all other components equal to 0. Therefore,

$$\theta(w) = -\sum_{i=1}^{m} w_i + \sum_{j=1}^{m} \min_{1\leq i\leq m} \{c_{ij} + w_i\} .$$

Furthermore, a vector ξ whose ith component ξ_i is given below is a subgradient of θ at w:

$$\xi_i = \sum_{j=1}^{m} \overline{x}_{ij} - 1 , \qquad i = 1, ..., m .$$

Again, a greedy one-pass heuristic was used to derive an overestimate θ^0 for θ^*. This time, for step size strategy 1, we used $r_1 = 5$ and $\epsilon_0 = 0.01$. The test problems used are the same as for the travelling salesman problem with $c_{ii} = \infty$ for $i = 1, ..., m$.

Table 2 gives the computational results. Comparing it with Table 1, one may note that similar remarks hold for this case as were made in Section 4.1.

References

[1] L.L. Barachet, Graphic solution of the travelling salesman problem, Operations Res. 5 (1957) 841–845.

[2] G.B. Dantzig, D.R. Fulkerson and S.M. Johnson, Solution of a large-scale travelling salesman problem, Operations Res. 2 (1954) 393–410.

[3] T.D. Edwards, An experimental investigation of subgradient optimization in mathematical programming, Unpublished Masters' Thesis, Georgia Institute of Technology (1979).

[4] M. Held and R.M. Karp, A dynamic programming approach to sequencing problems, J. Soc. Industr. Appl. Math. 10 (1962) 196–210.

[5] M. Held, P. Wolfe and H.D. Crowder, Validation of subgradient optimization, Math. Programming 6 (1974) 62–88.

[6] M. Held and R.M. Karp, The travelling salesman problem and minimum spanning trees, Operations Res. 18 (1970) 1138–1162.

[7] M. Held and R.M. Karp, The travelling salesman problem and minimum spanning trees: Part II, Math. Programming 1 (1971) 6–25.

[8] L.L. Karg and G.L. Thompson, A heuristic approach to solving travelling salesman problems, Management Sci. 10 (1964) 225–248.

[9] B.T. Poljak, A general method of solving extremum problems, Soviet Math. 8 (1967) 593–597.

[10] B.T. Poljak, Minimization of unsmooth functionals, U.S.S.R. Computational Math. and Math. Phys. 9 (1969) 14–29.

[11] R.T. Rockafellar, Convex Analysis (Princeton University Press, Princeton, NJ, 1970).