A MODEL FOR THE IMPLEMENTATION OF THE CONCEPTUAL OPTIMAL CONTROL ALGORITHM

Conf.dr.ing. Nicolae PAPUC

Faculty of Automation, Computers and Electronics, Automation Department, University of Craiova, E-mail: papy@automation.ucv.ro

Abstract. In this paper will shall consider algorithm for solving problems of the following form: Abstract problem: given a closed subset T of a Banach space **B**, construct points in T which have property P. In the case of nonlinear programming problems and discrete optimal control problems the space **B** will be $\Re^n$, while in the case of the continuous optimal control problems, the space **B** will be either L_2 or L_∞. Points with the property P will either be optimal for one of this optimization problems, or else they satisfy some optimality condition. Will call points in T with the property P, desirable. We shall always assume the T contains desirable points.

Key words: algorithm model, theorems of convergence, Banach space, desirable points.

The simplest algorithms for computing desirable points in a closed subset T of **B** utilize a **search function** $a: T \to T$ and a **stop rule** $c : T \to \Re^1$, and are of the following form:

1. Algorithm Model. $a: T \to T,\ c: T \to \Re^1$

Step 0. Compute a $z_0 \in T$.
Step 1. Set $i = 0$.
Step 2. Compute $a(z_i)$.
Step 3. Set $z_{i+1} = a(z_i)$.
Step 4. If $c(z_{i+1}) \ge c(z_i)$, stop; else, set $i = i+1$ and go to step 2.

We shall now show what one can hope to compute with such an algorithm.

2. Theorem. Suppose that

(i) $c(\cdot)$ is either continuous at all non-desirable points $z \in T$, or else $c(z)$ is bounded from below for $z \in T$;

(ii) for every $z \in T$; which is not desirable, there exists an $\varepsilon(z) > 0$ and a $\delta(z) < 0$ such that

3. $c(a(z')) - c(z') \le \delta(z) < 0$ for all $z' \in B(z, \varepsilon(z))$, where
$B(z,\varepsilon(z)) = \left\{ z \in T \middle| \, \|z' - z\|_{\mathbf{B}} \le \varepsilon(z) \right\}$

Then, either the sequence $\{z_i\}$ construct by algorithm **1** is finite and its next to last element is desirable, or else it is infinite and every accumulation point of $\{z_i\}$ is desirable.

Proof. We begin with the observation that – by negation – (ii) implies that if

4. $c(a(z)) \ge c(z)$

then z is desirable.

Now, suppose the sequence $\{z_i\}$ is finite, i.e., $\{z_i\} = \{z_0, z_1 ... z_k, z_{k+1}\}$. Then by step 4 $c(z_{k+1}) \ge c(z_k)$, and hence from **4**, z_k is desirable. Now suppose that the sequence $\{z_i\}$ is infinite, and that it has a subsequence which converges to the point z'. We express this as $z_i \to z'$ for $i \in K$, $K \subset \{0,1,2,...\}$. Assuming that z' is not desirable, there exist an $\varepsilon' > 0$, a $\delta' < 0$, and a $k \in K$ such that for all $i \ge k, i \in K$, ,

5. $\|z_i - z'\|_{\mathbf{B}} \le \varepsilon'$

and

6. $c(z_{i+1}) - c(z_i) \le \delta'$.

Hence, for any two consecutive points z_i, z_{i+j} of the subsequence, with $i \ge k$ (and $i, (i+j) \in K$) we must have

7.
$$\begin{aligned} c(z_{i+j}) - c(z_i) &= \left[c(z_{i+j}) - c(z_{i+j-1})\right] + \\ &+ \left[c(z_{i+j-1}) - c(z_{i+j-2})\right] + \dots \\ &+ \left[c(z_{i+1}) - c(z_i)\right] < c(z_{i+1}) - c(z_i) \le \delta' \end{aligned}$$

Now, for $i \in K$, the monotonically decreasing sequence $c(z_i)$ must converge either because $c(\cdot)$ is continuous at z' or else because $c(z)$ is bounded from below on T. But this is contradicted by **6**, which shows that the sequence $c(z_i)$ is not a Cauchy sequence for $i \in K$, and hence the theorem must be **true**.

A somewhat more sophisticated and useful model is obtained by substituting for the search function $a : T \to T$ of algorithm **1** a **set – valued** search function A mapping T into the set of all nonempty subsets of T (which also uses a stop rule $c : T \to \Re^1$).

8. **Algorithm Model.** $A : T \to 2^T$, $c : T \to \Re^1$

Step 0. Compute a $z_0 \in T$.
Step 1. Set $i = 0$.
Step 2. Compute a point $y \in A(z_i)$.
Step 3. Set $z_{i+1} = y$.
Step 4. If $c(z_{i+1}) \geq c(z_i)$, stop; else, set i = i+1 and go to step 2.

9. Theorem. Consider algorithm **8**. Suppose that

(i) $c(\cdot)$ is either continuous at all non-desirable points $z \in T$, or else c(z) is bounded from below for $z \in T$;

(ii) for every $z \in T$ which is not desirable, there exist an $\varepsilon(z) > 0$ and a $\delta(z) < 0$ such that.

10. $c(z'') - c(z') \leq \delta(z) < 0$

for all $z' \in T$ such that $\| z' - z \|_{\mathbf{B}} \leq \varepsilon(z)$, and for all $z'' \in A(z')$.

Then, either the sequence $\{z_i\}$ constructed by algorithm **8** is finite and its next to last element is desirable, or else it is infinite and every accumulation point of $\{z_i\}$ is desirable.

Proof. We begin with the observation that **(ii)** of **8** implies that if $c(z') \geq c(z)$ for at last one $z' \in A(z)$, then z is desirable. Now, the proof is similar to proof of the algorithm **1**.

Remark. The reader should be careful not to read more statements of the above convergence theorems than they actually say. Note that these theorems state only that *if* a convergent subsequence exists, then its limit point will be desirable. To ensure that accumulation points exist, it is necessary to make some additional assumptions. For example, one may assume that the set T is compact, or that the set $\{z \in T | \, c(z) \leq c(z_0)\}$ is compact where z_0 is the starting point for the algorithm. The reason for not including such assumptions in the statement of theorems such as **2** and **9** is that it is usually better to determine whether an algorithm will produce compact sequence by examining the algorithm in the light of the specific problem to which one wishes to apply it.

The convergence theorems **2** and **9** can be thought of as being extensions of Lyapunov's second method for the study of the stability of dynamical systems described by difference equations.

Algorithms **1** and **8** make sense if one can compute the point $z_{i+1} = a(z_i)$, or a point $z_{i+1} \in A(z_i)$, in a practically acceptable manner.

Now, one often invents algorithms of the form **1** or **8** in which the calculation of $a(z_i)$, or of points in the set $A(z_i)$, cannot be carried out in any practical manner. For example, the computation of such a point may require us to find the limit point of a sequence which we must first construct. As we have already mentioned, to resolve the practical difficulty this introduces, we must introduce suitable truncation, or approximation procedures. We shall now describe a few approximation procedures.

The most simple-minded approach to introducing a truncation procedure into algorithm **1** is to define an approximations set

11. $A_i(z) = \{ y \in T | \, \| y - a(z) \|_{\mathbf{B}} \leq \varepsilon \}$ where $\varepsilon \geq 0, z \in T$.

We can then modify **1** as follows:

Algorithm Model. Suppose that an $\varepsilon_0 > 0$ is given

Step 0. Compute a $z_0 \in T$.
Step 1. Set $i = 0$.
Step 2. Set $\varepsilon = \varepsilon_0$.
Step 3. Compute a $y \in A_i(z_i)$.
Step 4. If $c(y) - c(z_i) \leq -\varepsilon$, set $z_{i+1} = y$, set i=i+1 and go to step 2; else, set $\varepsilon = \varepsilon / 2$ and go to step 3.

Convention: In all algorithms, a sequence of commands such as "set" appearing in the instructions of a step, must be executed in the order in which they occur in the instructions.

The above algorithm will never stop after a finite number of iterations, since no stop commands are included. One of two things will occur. Either algorithm **12** will construct an infinite sequence $\{z_i\}$, or else it will jam up at a point z_k and cycle between steps 3 and 4, dividing ε by 2 at each cycle.

We shall now show what king of points one can compute using algorithm **12**.

13. Theorem. Suppose that the search function $a(\cdot)$ in **12** and the stop rule $c(\cdot)$ in step 4 of algorithm **12** satisfy condition **(ii)** of theorem **2** and that $c(\cdot)$ is uniformly continuous on T. Then either algorithm **12** jums up a desirable point after a finite number of iterations, or else it constructs an infinite sequence $\{z_i\}$ and every accumulation point of this sequence is desirable.

Proof. Suppose that the algorithm jammed up at the point z_k. Then it must be cycling between steps 3 and 4, producing in this process a sequence of vectors $y_j,\ j = 0,1,2,...$ in T such that $\|y_j - a(z_k)\|_{\mathbf{B}} \le \varepsilon_0 / 2^j$ and $c(y_j) - c(z_k) > -\varepsilon_0 / 2^j$. Hence, as $j \to \infty,\ y_j \to a(z_k)$, and since $c(\cdot)$ is assumed be continuous, we must have $c(a(z_k)) \ge c(z_k)$, i.e., z_k is desirable.

Now suppose that the sequence $\{z_i\}$ is infinite and that $z_i \to z'$ for $i \in K = \{0,1,2,...\}$, with z' a non-desirable point. Then there exists an integer $k \in K$ such that for $i \in K, i \ge k$,

14. $\|z_i - z'\|_{\mathbf{B}} \le \varepsilon', \quad \varepsilon' > 0$

and

15. $c(a(z_i)) - c(z_i) \le \delta' \quad \delta' < 0$

where ε' in **14** and δ' in **15** satisfy condition **(ii)** of theorem **2** with respect to the point z'. Now, since $c(\cdot)$ is uniformly continuous on T, there must exist an $\varepsilon'' > 0$ such that $c(y) - c(a(z)) \le -\delta'/2$ for all $z \in T$ and for all $y \in A_{\varepsilon''}(z)$, and hence we must have

16. $c(y) - c(z_i) \le \delta'/2$ for all $y \in A_{\varepsilon''}(z_i)$ with $i \in K,\ i \ge k$.

Let $\tilde{\varepsilon} = \min\{\varepsilon'', -\delta'/2\}$ and let $\tilde{\varepsilon} = \varepsilon_0 / 2^j$, where j is an integer such that

17. $\varepsilon_0 / 2^j \le \tilde{\varepsilon} \le \varepsilon_0 / 2^{j-1}$

Then, for every $i \in K, i \ge k$, and for every $y \in A_{\tilde{\varepsilon}}(z_i)$,

18. $c(y) - c(z_i) \ge -\tilde{\varepsilon}$

We therefore conclude that at every $z_i,\ i \in K,\ i \ge k$, the value of ε used in step 3 of **12** to compute the point z_{i+1} will be at least $\tilde{\varepsilon}$. Consequently, if i, i+j are two consecutive indices in K, with $i \ge k$, then

19. $c(z_{i+j}) - c(z_i) < \delta'/2 < 0$ (as in **6**)

Consequently, $c(z_i)$ cannot converge to $c(z')$ which contradicts our assumption that $c(\cdot)$ is continuous. Hence the theorem must be true.

By and large, to make sure that a point y is in $A_\varepsilon(z)$ may be almost as difficult as a compute a(z), a task we set out to avoid. Hence we are led the using approximations similar to $A_\varepsilon(z)$, but ones that do not require us to perform difficult tests. We now introduce in an algorithm model which generalizes **12** and which has a considerably broader range of applications. It constitutes a considerably more sophisticated approach to the implementation of conceptual algorithms, as will become clear from optimal control.

Note that we abuse notation in the use of the symbol A. The A in **20** is not the same as the A in **8** and **12**.

20. Algorithm Model.

$A : R^+ \times T \to 2^T,\quad c : T \to \Re^1,\ \varepsilon_0 > 0,\ \varepsilon' \in (0, \varepsilon_0)$.

Step 0. Compute a $z_0 \in T$.
Step 1. Set $i = 0$.
Step 2. Set $\varepsilon = \varepsilon_0$.
Step 3. Compute a $y \in A(\varepsilon, z_i)$.
Step 4. If $c(y) - c(z_i) < -\varepsilon$, set $z_{i+1} = y$, set i = i+1, and go to step 2.
Step 5. If $\varepsilon \le \varepsilon'$, perform a test to determine if z_i is desirable and go to step 6; else, set $\varepsilon = \varepsilon/2$ and go to step 3.
Step 6. If z_i is desirable, set $z_{i+1} = z_i$ and stop; else, set $\varepsilon = \varepsilon/2$ and go to step 3.

21. Theorem. Consider algorithm **20**. Suppose that

(i) $c(\cdot)$ is either continuous at all non-desirable points $z \in T$, or else c(z) is bounded from below for $z \in T$.

(ii) for every $z \in T$ which is not desirable, there exist an $\varepsilon(z) > 0$, a $\delta(z) < 0$, and $\gamma(z) > 0$, such that

22. $c(z'') - c(z') \le \delta(z) < 0$ for all $z' \in B(z, \varepsilon(z))$, for all $z'' \in A(y, z')$, for all $\gamma \in (0, \gamma(z)]$, where $B(z, \varepsilon(z)) = \{z' \in T \mid \|z' - z\|_{\mathbf{B}} \le \varepsilon(z)\}$

Then either the sequence $\{z_i\}$ constructed by algorithm **20** is finite and its last element is desirable, or else it is infinite and every accumulation point of $\{z_i\}$ is desirable.

Proof. First we must show that algorithm **20** is well-defined, i.e., that it cannot jam up at a non-desirable point z_k (It is quite clear that is cannot jam up at a desirable point without the stop command in step 6 being executed). Thus, suppose that the algorithm jams up at z_k, a non-desirable point. Then the algorithm must be producing an infinite sequence of vectors $y_j \in A(\varepsilon_0 / 2^j, z_k),\ j = 0,1,2,...,$ such that $c(y_j) - c(z_k) > -\varepsilon_0 / 2^j$, which prevents the construction of z_{k+1}. However, by assumption **(ii)**, since z_k is not desirable, there exist an $\varepsilon(z_k) > 0$, a $\delta(z_k) < 0$, and a $\gamma(z_k) > 0$ for which **22** holds. Also, there exists an integer $j' > 0$ such that $\varepsilon_0 / 2^{j'} \le \min\{-\delta(z_k, \delta(z_k)\}$, and hence, from **22**, we must have $c(y_j) - c(z_k) \le \delta(z_k) \le -\varepsilon_0 / 2^j$ for all $j \ge j'$, which contradicts our hypothesis that the algorithm jammed up at z_k. Consequently, algorithm **20** is well-defined.

Now, since the construction of a new point z_{i+1} can only stop when the stop command in step 6 of **20** is executed, the case of the sequence $\{z_i\}$ being finite is trivial. Hence, let us suppose that the sequence $\{z_i\}$ is infinite and that $z_i \to z'$, for $i \in K \subset \{0,1,2,...\}$, where the accumulation point z' is not desirable. Then, by **(ii)**, there exist an $\varepsilon(z') > 0$, a $\delta(z') < 0$, and a $\gamma(z') > 0$ for which **22** holds. Since $z_i \to z'$ for $i \in K$, there must exist a $k' \in K$ such that $z_i \in B(z', \varepsilon(z'))$ for all $i \in K$, $i \ge k'$, and there is also an integer k such that $\varepsilon_0 / 2^k \le \min\{-\delta(z'), \gamma(z')\}$. Hence, for any two consecutive points z_i, z_{i+j}, $i \ge k'$, of the subsequence $\{z_i\}_{i \in K}$, we must have

23. $$c(z_{i+j}) - c(z_i) = [c(z_{i+j}) - c(z_{i+j-1})] + ... + $$
$$+ [c(z_{i+1}) - c(z_i)] < c(z_{i+1}) - c(z_i) \le -\varepsilon_0 / 2^k.$$

which shows the sequence $c(z_i)$, $i \in K$, is not Cauchy (i.e., it does not converge). However, $c(z_i)$, $i \in K$, must converge because of assumption **(i)** and hence we get a contradiction.

Remark. In some of the algorithm it is easy to determine whether z_i is desirable in a process of calculating a $y \in A(\varepsilon, z_i)$. In such cases step 5 of **20** can be eliminated altogether, provided that the test for the desirability of z_i together with the corresponding stop command are incorporated into an expanded version of step 3. It should have no difficulty in making a mental adjustment for variations of the kind.

Under the assumptions of theorem **21**, the algorithm below has the same convergence properties as algorithm **20**.

24. Algorithm Model.

$A : \Re^+ \times T \to 2^T$, $c : T \to \Re^1$, $\varepsilon_0 > 0$, $\varepsilon' \in (0, \varepsilon_0)$, $\alpha > 0$, $\beta \in (0,1)$

```
Step 0. Compute a $z_0 \in T$.
Step 1. Set $i = 0$.
Step 2. Set $\varepsilon = \varepsilon_0$.
Step 3. Compute a $y \in A(\varepsilon, z_i)$.
Step 4. If $c(y) - c(z_i) \le -\alpha\varepsilon$, set $z_{i+1} = y$, set $i = i + 1$, and go to step 2; else, go to step 5.
Step 5. If $\varepsilon \le \varepsilon'$, perform a test to determine if $z_i$ is desirable and go to step 6;else,set $\varepsilon = \beta\varepsilon$ and go to step 3.
Step 6. If $z_i$ is desirable, set $z_{i+1} = z_i$ and stop; else, set $\varepsilon = \beta\varepsilon$ and go to step 3.
```

In algorithm model **20** and an in its generalization **24**, we begin each iteration with $\varepsilon = \varepsilon_0$. In some situation, this may prove to be inefficient, since, as z_i approaches a desirable point $\hat{z}$, we may be spending too much time in the loop which decreases ε to an acceptable value. When this is known to be the case, all we need to do is it change the model **24** slightly, as shown below, to make it time-variant.

25. Algorithm Model.

$A : \Re^+ \times T \to 2^T$, $c : T \to \Re^1$, $\tilde{\varepsilon}_0 > 0$, $\varepsilon' \in (0, \tilde{\varepsilon}_0)$, $\alpha > 0$, $\beta \in (0,1)$

```
Step 0. Compute a $z_0 \in T$.
Step 1. Set i = 0.
Step 2. Set $\varepsilon = \tilde{\varepsilon}_0$.
Step 3. Compute a $y \in A(\varepsilon, z_i)$.
Step 4. If $c(y) - c(z_i) \le -\alpha\varepsilon$, set $z_{i+1} = y$, set i=i+1, and go to step 3; else, go to step 5.
```

Comment. Algorithm **25** is time-varying, because from step 4 we return not to step 2, as in **24**, but step 3.

```
Step 5. If $\varepsilon \le \varepsilon'$, perform a test to determine if $z_i$ is desirable and go to step 6; else, set $\varepsilon = \beta\varepsilon$, and go to step 3.
Step 6. If $z_i$ is desirable, set $z_{i+1} = z_i$ and stop; else, set $\varepsilon = \beta\varepsilon$ and go to step 3.
```

REFERENCES

[1]. Bellman, R., and Kalaba, R., Quasilinearization and Boundary Value Problems, Elsevier, New York, 1965.
[2] Călin, S., Belea, C., Sisteme automate şi optimale, Ed. Tehnică, Bucureşti, 1971.
[3] Papuc, N.M., Dezvoltarea şi aplicarea unor concepte de optimizare multilocală la proiectarea şi conducerea sistemelor automate – Teza de doctorat, Universitatea din Craiova, 1983.
[4] Polak, E., Computational Methods in Optimization, Academic Press, N.Y. and London, 1971.
[5] Sima, V., Varga, A., Practica optimizării asistată de calculator, Ed. Tehnică, Bucureşti, 1986.