Proc. Estonian Acad. Sci. Engng., 1997, 3, 3, 196–204
https://doi.org/10.3176/eng.1997.3.05

APPLICATION OF ITERATIVE DYNAMIC PROGRAMMING TO OPTIMAL CONTROL OF NONSEPARABLE PROBLEMS*

Rein LÜÜS

Department of Chemical Engineering, University of Toronto, 200 College St., Toronto, Ontario, Canada M5S 1A4

Received 26 September 1996, revised 10 March 1997, accepted 15 May 1997

Abstract. A nonseparable optimization problem was used to compare the computational efficiency of iterative dynamic programming (IDP) to direct search optimization. As the number of stages was increased, it was found that the advantages of IDP become more remarkable. To solve a 100-stage problem with 3 state variables and 3 control variables is much more efficient by the use of IDP than by direct search optimization.

Key words: nonseparable optimization, iterative dynamic programming.

1. INTRODUCTION

The requirements of low dimensionality and separability of the optimization problem have limited the use of dynamic programming to solving only simple optimal control problems [1]. However, by using dynamic programming in an iterative manner, Lüüs [2–4] showed that the dimensionality problem can be overcome. In fact, the iterative dynamic programming (IDP) has been used successfully for establishing optimal control of a system described by 250 differential equations with 250 control variables [3], and for solving highly nonlinear systems encountered by chemical engineers [5]. Recently Lüüs and Tassone [6] showed that IDP can also be used successfully on the nonseparable problems studied by Li and Haimes [7] through the use of the best available values from the previous iteration for the required variables. On a higher dimensional

* Presented at the Congress of Estonian Scientists, 11–15 August 1966, Tallinn, Estonia.

problem, they showed that IDP is computationally equivalent to the direct search optimization method of Lüüs and Jaakola [8].

The purpose of this paper is to compare IDP to the direct search optimization procedure in solving the 3-dimensional nonseparable example used by Lüüs and Tassone [6]. Of special interest is to investigate how these optimization procedures compare when the number of stages becomes very large.

2. OPTIMIZATION PROBLEM

The problem we consider here is the system described by three nonlinear difference equations

$$x_1(k+1) = \frac{x_1(k)}{1+0.01u_1(k)(3+u_2(k))}, \tag{1}$$

$$x_2(k+1) = \frac{x_2(k)+u_1(k)x_1(k+1)}{1+u_1(k)(1+u_2(k))}, \tag{2}$$

$$x_3(k+1) = \frac{x_3(k)}{1+0.01u_2(k)(1+u_3(k))}, \tag{3}$$

where x is the state vector of dimension 3, k is the index pertaining to the stage number, u is the control vector of dimension 3, with the initial condition

$$x(0) = [2 \quad 5 \quad 7]^T. \tag{4}$$

The control variables are constrained by

$$0 \le u_1(k) \le 4, \tag{5}$$

$$0 \le u_2(k) \le 4, \tag{6}$$

$$0 \le u_3(k) \le 0.5. \tag{7}$$

The performance index I to be minimized is

$$I = x_1^2(P) + x_2^2(P) + x_3^2(P) + \left[\sum_{k=1}^{P} (x_1^2(k-1) + x_2^2(k-1) + 2u_3^2(k-1)) \sum_{k=1}^{P} (x_3^2(k-1) + 2u_1^2(k-1) + 2u_2^2(k-1)) \right]^{1/2}, \tag{8}$$

where P is the number of stages. Lüüs and Tassone [6] chose $P = 20$. Here, we will also investigate the cases when $P = 50$ and $P = 100$.

3. RESULTS

All computations were carried out in double precision on Pentium/166 PC, using the WATCOM FORTRAN compiler, version 9.5. For computations based on IDP, we took a single grid point and chose the candidates for the control variables at random inside the admissible region. The initial region size for every run was taken to be $r^{(0)} = [4\ 4\ 0.5]^T$, and the initial control policy was taken as $u^{(0)} = [2\ 2\ 0.25]^T$ at each stage. For IDP, we used a multipass method, each consisting of 30 iterations. After every iteration, the region was contracted by an amount $\gamma = 0.85$, and after every pass, the region was restored to $\eta = 0.5$ of its value at the beginning of the previous pass. In using direct search based on region contraction [8], we used the same starting conditions, but a region contraction factor of 0.95, which has been found to be a reasonable value for very complex systems [9]. The number of iterations was set at 201, but the number of search points was varied. For the cases $P = 50$ and $P = 100$, we used the direct search method in a multi-pass fashion to reduce computation time.

Case 1: $P = 20$

As expected, as the number of random points is increased, the answer obtained by direct search optimization is improved. As Table 1 shows, more than 2000 random points per iteration are required to obtain the optimum to five figures. The computation time of 81 s on the Pentium/166 PC for 3000 points per iteration is quite reasonable. Therefore, the use of direct search optimization for this problem with 20 stages (a total of 60 variables) appears to be quite satisfactory.

Table 1

Values of the performance index obtained by direct search optimization with $P = 20$

No. of random points per iteration	Performance index I	Computation time, s
100	209.41	2.8
500	209.29	13.6
1000	209.28	27.0
2000	209.28	54.0
3000	209.27	80.9
5000	209.27	134.9
10000	209.27	269.6

In solving this problem with IDP, we took a single grid point and allowed 5 passes of 30 iterations, each as described above. The convergence was very rapid, and computationally, the results were obtained considerably faster than with the use of the optimization by Lüüs and Jaakola [8]. Although there are three control variables, the use of only three allowable values for control gives convergence to the optimum $I = 209.27$ in less than one second of computation time. The convergence to the optimum is not monotonic, as shown in Fig. 1, where the oscillatory nature of convergence is observed. Since a log scale is used, the oscillations during the fifth pass appear to be very large, whereas in actual terms, they are quite small. As Fig. 2 shows, when 25 randomly chosen values for control are used instead of 3, there is more rapid approach to the origin in the first pass, but the end result after 5 passes is approximately the same. It is interesting to note that, in each case, a better value for the performance index was obtained at the end of the fourth pass (as seen from the starting point for the fifth pass) than at the end of the fifth pass. The computation time for 5 passes with $R = 25$ allowable values for control was 2.7 s, where R is the number of randomly chosen candidates for control.

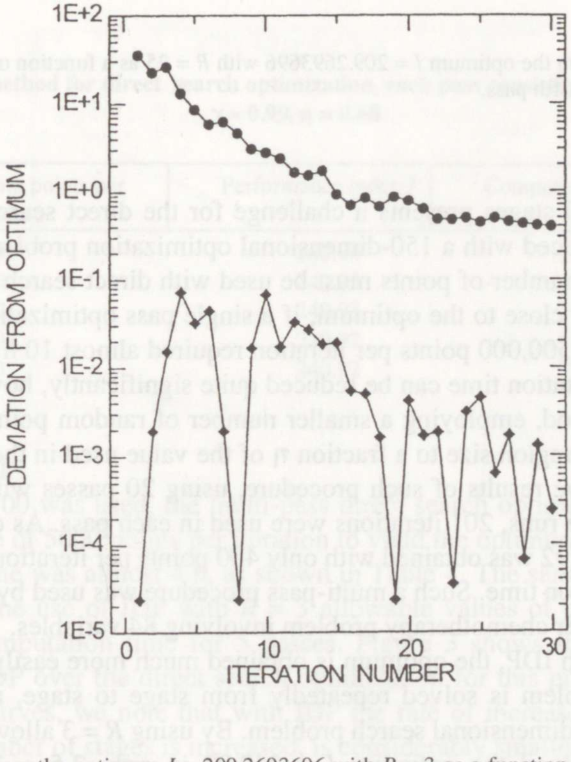

Fig. 1. Deviation from the optimum $I = 209.2693696$ with $R = 3$ as a function of iteration number; ●●● first pass, ◆◆◆ fifth pass.

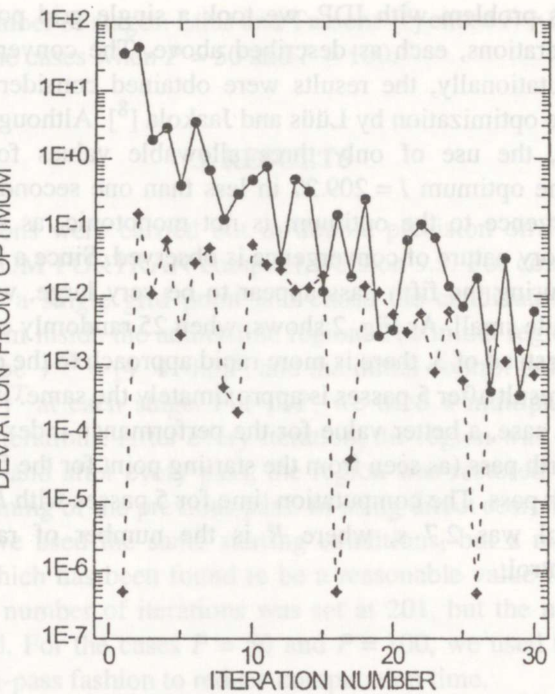

Fig. 2. Deviation from the optimum $I = 209.2693696$ with $R = 25$ as a function of iteration number; ●●● first pass, ◆◆◆ fifth pass.

Case 2: $P = 50$

The use of 50 stages presents a challenge for the direct search optimization, because we are faced with a 150-dimensional optimization problem. As shown in Table 2, a large number of points must be used with direct search optimization to obtain an answer close to the optimum, if a single pass optimization procedure is used. The use of 500,000 points per iteration required almost 10 h of computation time. The computation time can be reduced quite significantly, however, by using a multipass method, employing a smaller number of random points per iteration, but restoring the region size to a fraction η of the value used in the previous pass. Table 3 shows the results of such procedure, using 20 passes with $\gamma = 0.99$ and $\eta = 0.60$. In these runs, 201 iterations were used in each pass. As can be seen, the optimum $I = 240.92$ was obtained with only 400 points per iteration in less than 10 min of computation time. Such a multi-pass procedure was used by Lüüs et al. [10] in solving a cancer chemotherapy problem involving 84 variables.

However, with IDP, the optimum is obtained much more easily, since a three-dimensional problem is solved repeatedly from stage to stage, and we are not faced with a 150 dimensional search problem. By using $R = 3$ allowable values for control, we obtained the optimum $I = 240.92$ in only 2.5 s. Therefore, IDP becomes considerably more attractive than direct search optimization when the number of stages is large.

Table 2

Values of the performance index with $P = 50$, obtained by direct search optimization, using a single pass of 201 iterations with $\gamma = 0.95$

No. of random points per iteration	Performance index I	Computation time, s
100	244.72	6.9
500	242.61	34.4
1000	241.66	68.9
2000	241.43	137.2
5000	241.14	342.8
10000	241.02	685.2
20000	240.98	1370.7
50000	240.94	3426.3
100000	240.93	6835.7
200000	240.93	13707.8
500000	240.92	34270.4

Table 3

Use of 20-pass method for direct search optimization, each pass consisting of 201 iterations; $\gamma = 0.99$, $\eta = 0.60$

No. of random points per iteration	Performance index I	Computation time, s
100	240.94	139.0
200	240.93	277.2
300	240.93	415.4
400	240.92	553.7
500	240.92	692.2

Case 3: $P = 100$

When $P = 100$ was used, the multi-pass direct search optimization procedure required the use of 5000 points per iteration to yield the optimum $I = 258.34$. The computation time was almost 4 h, as shown in Table 4. The same value of I was obtained with the use of IDP with $R = 3$ allowable values of control, requiring only 7 s of computation time for 5 passes. Figure 3 shows the computational advantage of IDP over the direct search optimization for this problem. From the slopes of the curves, we note that with IDP the rate of increase in computation time, as the number of stages is increased, is considerably smaller than with direct search optimization.

Table 4

Use of 20-pass method for direct search optimization with $P = 100$, with each pass consisting of 201 iterations; $\gamma = 0.99$, $\eta = 0.60$

No. of random points per iteration	Performance index I	Computation time, s
100	258.42	290.3
200	258.38	578.8
300	258.36	867.6
400	258.36	1156.6
500	258.36	1445.8
600	258.35	1734.4
800	258.35	2312.5
1000	258.35	2890.3
2000	258.35	5676.3
3000	258.35	8514.8
4000	258.35	11352.4
5000	258.34	14190.3

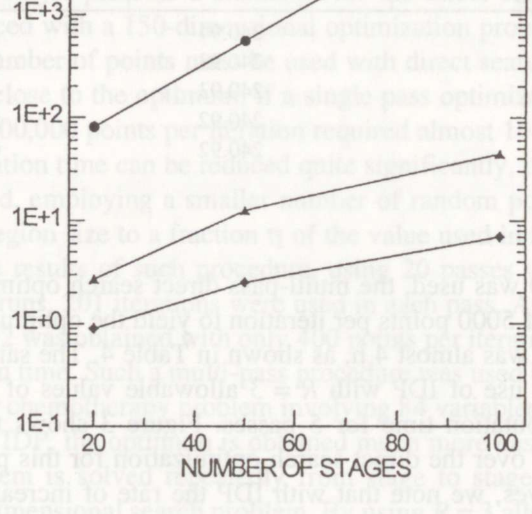

Fig. 3. Computation time as a function of the number of stages; ●●● direct search optimization, ▲▲▲ IDP with $R = 25$, ◆◆◆ IDP with $R = 3$.

4. DISCUSSION AND CONCLUSIONS

Since IDP is considerably faster than direct search optimization in establishing the optimal control of systems with a large number of stages, it is worthwhile to try that method even when the problem is nonseparable. At present, it is not quite clear how IDP can be used effectively for complex recycle systems, since the recycle stream at the outlet must be equal to the recycle stream at the inlet. One possible approach to force this equality is to use penalty functions. Research in this area is continuing. In general, however, to use values that have been calculated from the previous iteration for those that are not available at the current iteration appears to work very well for many nonseparable problems.

ACKNOWLEDGEMENT

Financial support from the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

REFERENCES

1. Bellman, R. E. and Dreyfus, S. E. *Applied Dynamic Programming*. Princeton University Press, Princeton, N.J., 1962.
2. Lüüs, R. Optimal control by dynamic programming using accessible grid points and region reduction. *Hungarian J. Ind. Chem.*, 1989, **17**, 523–543.
3. Lüüs, R. Application of dynamic programming to high-dimensional nonlinear optimal control problems. *Int. J. Cont.*, 1990, **52**, 239–250.
4. Lüüs, R. Numerical convergence properties of iterative dynamic programming when applied to high dimensional systems. *Trans. IChemE*, 1996, **74**, 55–62.
5. Bojkov, B. and Lüüs, R. Optimal control of nonlinear systems with unspecified final times. *Chem. Eng. Sci.*, 1996, **51**, 905–919.
6. Lüüs, R. and Tassone, V. Optimal control of nonseparable problems by iterative dynamic programming. In *Proc. 42nd Canadian Chem. Eng. Conf.*, Toronto, Canada, October 18–21, 1992, 81–82.
7. Li, D. and Haimes, Y. Y. New approach to nonseparable dynamic programming problems. *JOTA*, 1990, **64**, 311–330.
8. Lüüs, R. and Jaakola, T. H. I. Optimization by direct search and systematic reduction of the size of search region. *AIChE J.*, 1973, **19**, 760–766.
9. Spaans, R. and Lüüs, R. Importance of search-domain reduction in random optimization. *JOTA*, 1992, **73**, 635–638.
10. Lüüs, R., Hartig, F., and Keil, F. J. Optimal drug scheduling of cancer chemotherapy by direct search optimization. *Hungarian J. Ind. Chem.*, 1995, **23**, 55–58.

ITERATIIVSE DÜNAAMILISE PLANEERIMISE RAKENDAMINE MITTEERALDATAVATE ÜLESANNETE OPTIMAALSEKS JUHTIMISEKS

Rein LÜÜS

Iteratiivse dünaamilise planeerimise arvutusliku efektiivsuse võrdlemiseks otsese otsimisega optimeerimisprotseduuriga on vaadeldud mitteeraldatavat optimeerimisülesannet. On näidatud, et meetodi eelis on märkimisväärne süsteemi olekute arvu suurenemisel. Nii on 100-etapilise ülesande lahendamine kolme olekumuutuja ja kolme juhtmuutuja korral palju efektiivsem iteratiivse dünaamilise planeerimise abil.