



Optimization

Iranian Journal of Optimization
Volume 7, Issue 2, 2015, 823-829
Research Paper



Islamic Azad University
Rasht Branch
E-ISSN:2008-5427

Online version is available on: www.ijo.iaurasht.ac.ir

A Numerical Approach for Optimal Control Model of the Convex Semi-Infinite Programming

Hamid Rouhparvar*

Department of Mathematics, Saveh Branch, Islamic Azad University, Saveh, Iran.

Received: 08 April 2015

Accepted: 26 June 2015

Keywords:

Optimal control

iterative dynamic programming

neural networks

convex semi-infinite programming

Mathematics Subject Classification (2000): 90C25, 90C34, 65K99

Abstract

In this paper, convex semi-infinite programming is converted to an optimal control model of neural networks and the optimal control model is solved by iterative dynamic programming method. In final, numerical examples are provided for illustration of the purposed method.

*Correspondence E-mail: rouhparvar59@gmail.com

INTRODUCTION

Semi-infinite programming models apply in mathematics, engineering, physics, social and other sciences when some processes or systems depend on a finite dimensional variable and are described with the help of an infinite number of constraints. In the last decades, semi-infinite optimization has become a topic of a special interest due to a number of practical applications and the relationship with other mathematical fields (Hettich et al., 1993; Polak, 1983; Weber, 2002).

Many optimization problems are naturally cast as semi-infinite problems; for instance, continuous time, optimal control problems subject to all time state constraints. A study of explicit optimality conditions as well as a comparison of these conditions with the known optimality conditions of convex semi-infinite programming (CSIP) (Ben Tal et al., 1980; Hettich et al., 1993; Kortanek et al., 2005) is the subject of a separate investigation (Kostyukova et al., 2006).

In (Song et al., 1998), the mechanism of the optimization neural networks is studied from the point of view of control systems. It is shown that an optimization neural network can be modeled as an optimal control problem. We show how CSIP by using of method told in (Song et al., 1998) will transform to an optimal control problem and then it is solved by iterative dynamic programming (IDP) method. The initial ideas on IDP were developed and tested by (Luus, 1990) and then refined (Luus, 1989) to make the computational procedure much more efficient. IDP method provided a very convenient way of investigating the effect of the choice of the final time in optimal control problems (Luus, 1991), also numerical convergence properties of that can see in (Lin et al., 1998; Luus, 2000; Luus, 1996). We show how the IDP method was applied to solve this specific problem.

CONVEX SEMI-INFINITE PROGRAMMING Formulation problem

In this section we consider the mathematical program

$$\begin{aligned} \min f(x) \\ \text{s.t. } g_i(x, s) \leq b_i(s), i = 1, \dots, m \\ x \in \mathbb{R}^n, s \in [a, b], \end{aligned} \quad (1)$$

where f is a convex function and $g_i(x, s)$, ($i=1, \dots, m$) are concave functions with respect to the first argument. Under these assumptions (1) is a CSIP. It is also assumed that f, g_i, b_i ($i=1, \dots, m$) are twice continuously differentiable.

Transformation of CSIP to optimal control problem

Now problem (1) transform to an optimal control model. If consider Hopfield network or Kennedy and Chua's network in state linear (Kennedy et al., 1987; Maa et al., 1992; Song et al., 1998) thus the Hopfield and the Kennedy and Chua's networks are equivalent to state feedback control systems. Now, the mechanism of those optimization neural networks can be formulated as follows: The optimization neural networks are equivalent to state feedback control systems; the dynamics of the system is determined by the objective function; the control action exists only when constraint violation takes place and the control signal is determined by the magnitude of the violations. This can be shown true for the general case.

In general, if $f(x)$ is non-linear and if the penalty method (Luenerger, 1984) is applied to solve (1), then we can obtain an unconstrained optimization problem

$$\min P(x) = f(x) + \frac{k}{2} \sum_{i=1}^m (h_i^+(x))^2 \quad (1)$$

where k is a positive number and

$$\begin{aligned} h_i(x) &= g_i(x, s) - b_i(s), \\ h_i^+(x) &= \max\{0, \max_{s \in [a, b]} h_i(x)\}, \quad i = 1, \dots, m. \end{aligned}$$

Let us define the control system

$$\dot{x} = - \frac{\partial f(x)}{\partial x} + u \quad (3)$$

where $u \in \mathbb{R}^n$ is an control variable and $\partial f(x)/\partial x = (\partial f/\partial x_1, \partial f/\partial x_2, \dots, \partial f/\partial x_n)^T$.

Proposition 2.1. If for any k , (2) has an optimal solution, and if for system (3) we can find a state

feedback control $u = -k \sum_{i=1}^m h_i^+(x) \frac{\partial h_i(x)}{\partial x}$ such that the closed-loop system

$$\dot{x} = -\frac{\partial f(x)}{\partial x} - k \sum_{i=1}^m h_i^+(x) \frac{\partial h_i(x)}{\partial x} \quad (2)$$

is asymptotically stable at x^* , then the optimal solution to (2) will be the equilibrium state of (4).

Note: $\frac{\partial h_i(x)}{\partial x} = \left(\frac{\partial h_i}{\partial x_1}, \frac{\partial h_i}{\partial x_2}, \dots, \frac{\partial h_i}{\partial x_n} \right)^T$.

Proof. See (Song et al., 1998).

Under the combined action of both $h_i^+(x)$ and $\partial h_i(x)/\partial x$, the system is controlled to reach its equilibrium. Thus, the mechanism of the optimization neural networks can be in general explained with control theory.

Because of its specific roles, we refer to $u = -k \sum_{i=1}^m h_i^+(x) \frac{\partial h_i(x)}{\partial x}$ as the penalty control law. We can show that under the penalty control, $P(x)$ of (2) is also a Lyapunov function of (3).

Proposition 2.2. Under the penalty control, $P(x)$ of (2) is a Lyapunov function of system (3).

Proof. See (Song et al., 1998).

From Proposition 2.2, it can be seen that under the penalty control, system (3) evolves along a direction such that the Lyapunov function $P(x)$ decreases the most. The penalty control law can be obtained if we take the optimization problem as an optimal control problem.

In solving optimization problems the use of neural networks, generally the steady state of the network is taken as the solution. So, letting x in (2) be $x(t_f)$ (generally $t_f = \infty$) gives us

$$\begin{aligned} \min J(u) &= f(x(t_f)) + \frac{k}{2} \sum_{i=1}^m (h_i^+(x(t_f)))^2 \\ \text{s.t. } \dot{x}(t) &= -\frac{\partial f(x(t))}{\partial x(t)} + u(t) \end{aligned} \quad (3)$$

Now the problem is to find an optimal state feedback control $u^*(t)$ such that under $u^*(t)$ when system (3) reaches an equilibrium state, (5) is

minimal. This can be expressed as a standard terminal control problem:

$$\begin{aligned} \min J(u) &= f(x(t_f)) + \frac{k}{2} \sum_{i=1}^m (h_i^+(x(t_f)))^2 \\ \text{s.t. } \dot{x}(t) &= -\frac{\partial f(x(t))}{\partial x(t)} + u(t) \end{aligned} \quad (4)$$

where $x(0)$ and t_f is given. Under certain conditions, the penalty control is also the optimal state feedback control, see (Song et al., 1998).

Transformation of optimal control problem to finite-horizon

In this section, by suitable change variable, we transform the interval $[0, \infty)$ to $[0, 1)$, and then obtain optimal control and the corresponding trajectory in this interval. By suitable change variable of the form:

$$\theta = \frac{2}{\pi} \text{Arctan}(t) \quad \text{or} \quad t = \tan\left(\frac{\pi}{2}\theta\right),$$

the problem (6) is transformed to the following variational optimal control problem,

$$\begin{aligned} \min J(u) &= f(x(\tan(\frac{\pi}{2}\theta_f))) + \frac{k}{2} \sum_{i=1}^m (h_i^+(x(\tan(\frac{\pi}{2}\theta_f))))^2 \\ \text{s.t. } x(\tan(\frac{\pi}{2}\theta)) &= -\frac{\partial f(x(\tan(\frac{\pi}{2}\theta)))}{\partial x(\tan(\frac{\pi}{2}\theta))} + u(\tan(\frac{\pi}{2}\theta)) \\ \theta_f &= 1, \theta \in [0, \theta_f). \end{aligned}$$

Assume $y(\theta) = x\left(\tan\left(\frac{\pi}{2}\theta\right)\right)$ and $v(\theta) = u\left(\tan\left(\frac{\pi}{2}\theta\right)\right)$, then we have the following variational problem,

$$\begin{aligned} \min J(v) &= f(y(\theta_f)) + \frac{k}{2} \sum_{i=1}^m (h_i^+(y(\theta_f)))^2 \\ \text{s.t. } y(\theta) &= \frac{\pi}{2} \sec^2\left(\frac{\pi}{2}\theta\right) \left(-\frac{\partial f(y(\theta))}{\partial y(\theta)} + v(\theta) \right) \\ \theta_f &= 1, \theta \in [0, \theta_f). \end{aligned} \quad (5)$$

where $Y(0)=x(0)$ is given and $y(1)=y^*=x^*$ is optimal solution of CSIP.

ITERATIVE DYNAMIC PROGRAMMING

To set up the problem into a sequence of stages, as required for dynamic programming, we approximate the optimal control problem by requiring a piecewise constant control policy over P stages, each of length L , so that ($\theta_0=0$)

$$L = \theta_f / P \quad (6)$$

then the performance index in terminal time is approximated by

$$J = f(v(\theta_f)) + \frac{k}{2} \sum_{i=1}^m (h_i^+(v(\theta_f)))^2. \quad (7)$$

Note: $v(k-1)$ is the constant control in the time interval $\theta_{k-1} \leq \theta \leq \theta_k (k=1, \dots, P)$.

We can consider the system at these P stages. The following steps were used:

1. Divide the time interval $[0, 1)$ into P time stages, each of length L . At each time stage, we seek a constant value for the control vector v .

2. Choose the number of test values for the control vector v denoted by R , an initial control policy and the initial region size r_{in} ; also choose the region contraction factor γ used after every iteration and the number of grid points N at each time stage.

3. Choose the total number of iterations and set the iteration number index to $j=1$.

4. Set the region size vector $r^j = r_{in}$.

5. By using the best control policy (the initial control policy for the iteration) integrate the equations, from $\theta=0$ to 1, N times with different values for control. This will generate N y -trajectories which provide the grid points. Store the values of y at the beginning of each time stage, so that $y(k-1)$ corresponds to the value of y at beginning of stage k .

6. Starting at stage P , corresponding to time $1-L$, for each of the N stored values for $y(P-1)$ from step 5 (grid points) integrate the differential equations from $1-L$ to 1, with each of the R allowable values for the control vector calculated from

$$V(P-1) = v^{*j}(P-1) + D r^j \quad (8)$$

where $v^{*j}(P-1)$ is the best value obtained in the previous iteration and D is a diagonal matrix of

different random numbers between -1 and 1. Out of the R values for the augmented performance index, choose the control values that give the minimum value, and store these values as $v(P-1)$. We now have the best control for each of these N grid points.

7. Step back to stage $P-1$, corresponding to time $1-2L$, and for each of the N grid points do the following calculations. Choose R values for $v(P-2)$ as in the previous step, and by taking as the initial state $y(P-2)$ integrate the differential equations over one stage length. Continue integration over the last time stage by using the stored value of $v(P-1)$ from step 6 corresponding to the grid point that is closest to the value of the state vector that has been reached. Compare the values of the performance index and store $v(P-2)$ that gives the minimum value for the augmented performance index.

8. Continue the procedure until stage 1, corresponding to the initial time $\theta=0$ and the given initial state, is reached. This stage has only a single grid point, since the initial state is specified. As before, integrate the differential equations and compare the R values of the augmented performance index and store the control $v(0)$ that gives the minimum augmented performance index. Store also the corresponding y -trajectory. This completes one iteration.

9. Reduce the region for allowable control

$$r^{j+1} = \gamma r^j \quad (9)$$

where j is the iteration number index. Use the best control policy from step 8 as the midpoint for the allowable values for the control denoted by the superscript.

10. Increment the iteration index j by 1 and go to step 5 and continue the procedure for the specified number of iterations.

NUMERICAL EXAMPLES

Example 4.1. Consider the following convex semi-infinite program

$$\begin{aligned} & \min 1 - x \\ & s.t. \quad x^2 - 1 \leq s^2 + s \\ & \quad \quad x \geq 0, s \in [0, 1]. \end{aligned}$$

Applying the method described in section 2.3, we have

$$\min J(v) = 1 - y(\theta_f) + \frac{k}{2} [\max\{0, \max_{s \in [0,1]} (-s^2 - s +$$

$$y(\theta_f)^2 - 1)\}]^2$$

$$\text{s.t. } \dot{y}(\theta) = \frac{\pi}{2} \sec^2\left(\frac{\pi}{2}\theta\right)(-1 + v(\theta))$$

$$y(\theta) \geq 0, \theta \in [0, \theta_f], \theta_f = 1.$$

Now we solve the above problem by using of IDP method. Suppose in IDP method, $y(0)=0$, $k=10^5$, $P=5$, $\gamma=0.75$, $N=3$, $R=4$ and $v(\cdot) \in [-0.44, 0.32]$, with 25 iteration we obtain result and optimal value objective is 0.0003 where this problem solved in (Luhandjula et al., 2001), and $x^*=1$. Graphs of the trajectory function and the piecewise constant control function are shown in Fig. 1-2. respectively.

Note: for constrain $y(\theta) \geq 0$ in IDP method see (Luus, 2000).

Example 4.2. Consider the following convex semi-infinite program

$$\min -4x_1 + x_2 + 3x_3 + \frac{x_1^2}{2} + \frac{x_2^2}{2} + \frac{x_3^2}{2} + \frac{x_4^2}{2}$$

$$\text{s.t. } -s^2x_1 + sx_2 + \sin(s)x_3 + x_4^2 \leq 0$$

$$s \in [-1, 2].$$

We have

$$\min J(v) = -4y_1(\theta_f) + y_2(\theta_f) + 3y_3(\theta_f) +$$

$$\frac{y_1^2(\theta_f)}{2} + \frac{y_2^2(\theta_f)}{2} + \frac{y_3^2(\theta_f)}{2} + \frac{y_4^2(\theta_f)}{2}$$

$$\text{s.t. } y_1(\theta) = \frac{\pi}{2} \sec^2\left(\frac{\pi}{2}\theta\right)(4 - y_1(\theta_f) + v_1(\theta))$$

$$y_2(\theta) = \frac{\pi}{2} \sec^2\left(\frac{\pi}{2}\theta\right)(-1 - y_2(\theta_f) + v_2(\theta))$$

$$y_3(\theta) = \frac{\pi}{2} \sec^2\left(\frac{\pi}{2}\theta\right)(-3 - y_3(\theta_f) + v_3(\theta))$$

$$y_4(\theta) = \frac{\pi}{2} \sec^2\left(\frac{\pi}{2}\theta\right)(-y_4(\theta_f) + v_4(\theta)),$$

$$\theta \in [0, \theta_f], \theta_f = 1.$$

Now with selection, $\theta \in [0, 1]$, $y_1(0)=y_2(0)=y_3(0)=0$, $y_4(0)=0.5$, $k=4 \times 10^8$, $P=7$, $\gamma=0.75$, $N=2$, $R=5$, $v_1(\cdot) \in [2, 2.8]$, $v_3(\cdot) \in [1.6, 2.1]$ and $v_4(\cdot) \in [-0.6,$

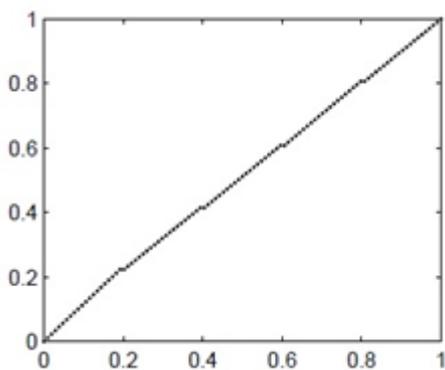


Fig. 1. Approximate trajectory $y(\cdot)$

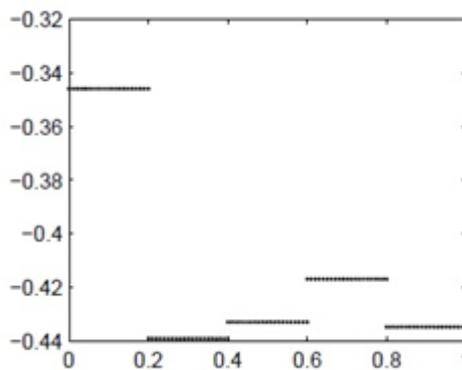


Fig. 2. The optimal control $v(\cdot)$

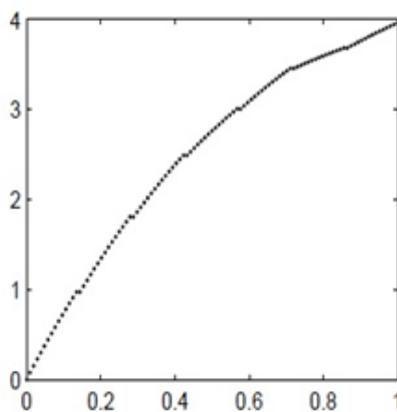


Fig. 3. Approximate trajectory $y_1(\cdot)$

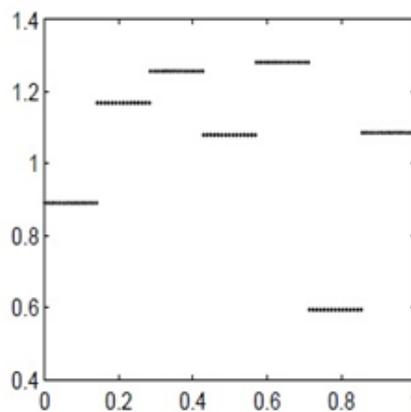


Fig. 4. The optimal control $v_2(\cdot)$

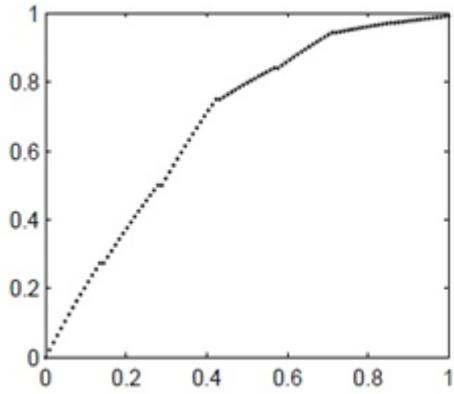


Fig. 5. Approximate trajectory $y_2(\cdot)$

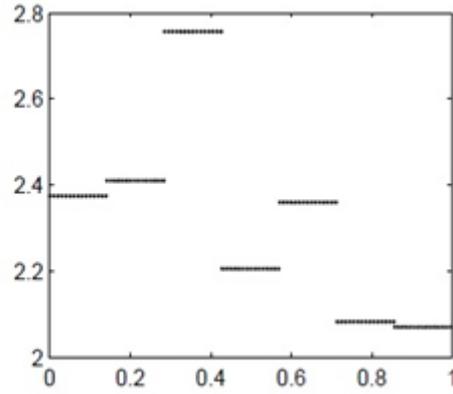


Fig. 6. The optimal control $v_2(\cdot)$

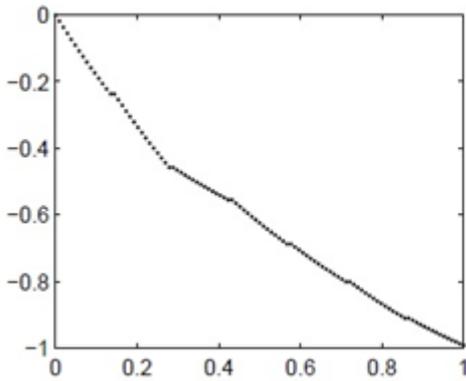


Fig. 7. Approximate trajectory $y_3(\cdot)$

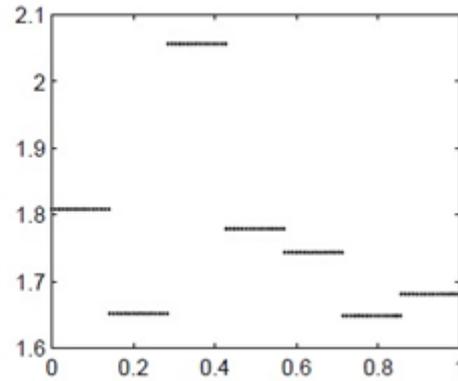


Fig. 8. The optimal control $v_3(\cdot)$

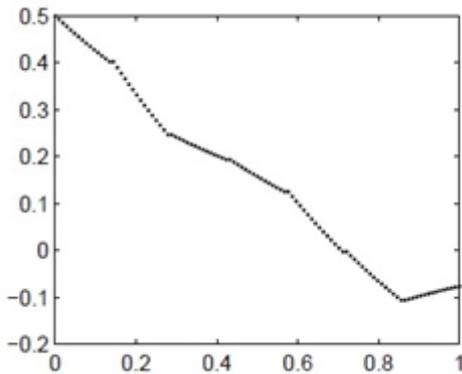


Fig. 9. Approximate trajectory $y_4(\cdot)$

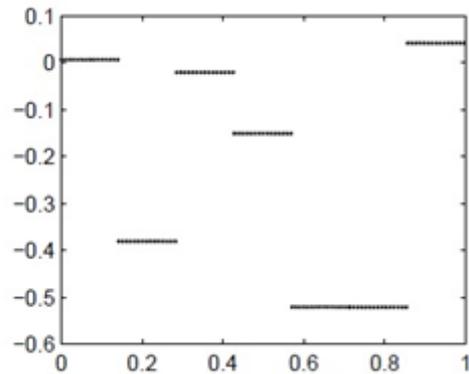


Fig. 10. The optimal control $v_4(\cdot)$

$0.1]$, with 25 iteration we obtain results

$$y_1^* = x_1^* = 3.9627, \quad y_2^* = x_2^* = 0.9907,$$

$$y_3^* = x_3^* = -0.9907, \quad y_4^* = x_4^* = -0.0771,$$

and optimal value objective is -8.9963, of course, this problem solved in (Kostyukova et al., 2005), where optimal solution is $x^* = (4, 1, -1, 0)$. Graphs of the trajectory functions and the piecewise constant control functions are shown in Fig. 3-10. respectively.

CONCLUSION

In this article, the mechanism was shown that convex semi-infinite programming using optimization neural networks could be modeled as optimal control problems. The penalty control law, determined by both the objective function and the violated constraints, played a key role in the process of optimization with neural networks. It was shown that the penalty control was also an optimal control. With process told, the algorithms developed in solving optimal control problems

especially IDP method can be used. But advantage of IDP method is computation $\sum_{i=1}^m h_i^+(y(\theta_f))^2$ in objective function during process of execute algorithm where after the amount numerical of $y(\theta_f)$ was determined in each iterate is computed with respect to parameter s easily.

ACKNOWLEDGEMENTS

The author would like to appreciate the anonymous reviewers for their helpful comments

REFERENCE

- Ben-Tal, A., Kerzner, L., & Zlobec, S. (1980). Optimality conditions for convex semi-infinite programming problems. *Naval Research Logistics (NRL)*, 27(3), 413-435.
- Hettich, R., & Kortanek, K. O. (1993). Semi-infinite programming: theory, methods, and applications. *SIAM review*, 35(3), 380-429.
- Hettich, R., & Still, G. (1995). Second order optimality conditions for generalized semi-infinite programming problems. *Optimization*, 34(3), 195-211.
- Kennedy, M., & Chua, L. (1987). Unifying the Tank and Hopfield linear programming circuit and the canonical nonlinear programming circuit of Chua and Lin. *IEEE Transactions on Circuits and Systems*, 34(2), 210-214.
- Kortanek, K.O., & Medvedev, V.G. (2005). Semi-infinite programming and new applications in finance. *Encyclopedia of Optimization*. Kluwer Academic, Dordrecht.
- Kostyukova, O. I., & Tchemisova, T. V. (2006). Convex semi-infinite programming: explicit optimality conditions.
- Kostyukova, O.I., Tchemisova, T.V., & Yermalinskaya, S. A. (2010). Convex semi-infinite programming: implicit optimality criterion based on the concept of immobile indices. *Journal of optimization theory and applications*, 145(2), 325-342.
- Lin, J. S., & Hwang, C. (1998). Enhancement of the global convergence of using iterative dynamic programming to solve optimal control problems. *Industrial & engineering chemistry research*, 37(6), 2469-2478.
- Luenberger, D. G. (1984). Linear and nonlinear programming. Reading: Addison-Wesley.
- Luhandjula, M. K., & Ouanes, M. (2001). A cutting-plane approach for semi-infinite mathematical programming. *African Journal of Science and Technology*, 2(1).1-10.
- Luus, R. (2000). Iterative dynamic programming. Monographs and surveys in pure and applied mathematics.
- Luus, R. (1991). Effect of the choice of final time in optimal control of nonlinear systems. *The Canadian Journal of Chemical Engineering*, 69(1), 144-151.
- Luus, R. (1996). Numerical Convergence Properties of Iterative Dynamic-Programming When Applied to High-Dimensional Systems. *Chemical engineering research & design*, 74(1), 55-62.
- Luus, R. (1989). Optimal control by dynamic programming using accessible grid points and region reduction. *Hungarian Journal of Industrial Chemistry*, 17(4), 523-543.
- Luus, R. (1990). Optimal control by dynamic programming using systematic reduction in grid size. *International Journal of Control*, 51(5), 995-1013.
- Maa, C. Y., & Shanblatt, M. A. (1992). Linear and quadratic programming neural network analysis. *IEEE transactions on neural networks*, 3(4), 580-594.
- Polak, E. (1983). Semi-infinite optimization in engineering design. In *Semi-infinite programming and applications* (pp. 236-248). Springer Berlin Heidelberg.
- Song, Q., & Leland, R.P. (1998). An optimal control model of neural networks for constrained optimization problems. *Optimal Control Applications and Methods*, 19(5), 371-376.
- Weber, G. W. (2002). Generalized semi-infinite optimization: theory and applications in optimal control and discrete optimization. *Journal of Statistics and Management Systems*, 5(1-3), 359-388.