

یک شبکه عصبی اصلاح شده کارا برای حل مسایل برنامه‌ریزی غیرخطی با محدودیت‌های هیبریدی

علیرضا ناظمی^{۱*}، سمیرا سوخت سرایی^۲، هرضیه مرتضایی^۳

۱- دانشیار، دانشگاه صنعتی شهرود، گروه ریاضی کاربردی، شهرود، ایران

۲- دانش آموخته کارشناسی ارشد، دانشگاه صنعتی شهرود، گروه ریاضی کاربردی، شهرود، ایران

۳- دانشجوی دکتری، دانشگاه صنعتی شهرود، گروه ریاضی کاربردی، شهرود، ایران

رسید مقاله: ۱۷ بهمن ۱۳۹۶

پذیرش مقاله: ۲۴ اسفند ۱۳۹۷

چکیده

یک مدل شبکه عصبی کارا برای حل یک مساله بهینه‌سازی غیرخطی با محدودیت‌های هیبریدی ارایه می‌شود. بر مبنای شرایط بهینگی کاروش-کان-تاکر و مفاهیمی از آنالیز محاسب و بهینه‌سازی، ابتدا یک مدل شبکه ارایه می‌شود. ثابت می‌شود که مدل شبکه عصبی ارایه شده پایدار لیپانوف بوده و همگرای سراسری به جواب مساله بهینه‌سازی اصلی می‌باشد. با ارایه چندین مثال عددی، کارایی مدل ارایه شده برای حل مساله بهینه‌سازی با محدودیت‌های هیبریدی نشان داده می‌شود.

کلمات کلیدی: برنامه‌ریزی محاسب، پایدار لیپانوف، محدودیت‌های هیبریدی، شبکه عصبی، همگرای سراسری.

۱ مقدمه

مسایل بهینه‌سازی درجه دوم در طیف وسیعی از کاربردهای علمی و مهندسی از جمله پردازش سیگنال، شناسایی سیستم، طراحی صافی، تقریب توابع، تجزیه و تحلیل رگرسیون و... به کار برده می‌شود [۱]-[۸]. در بسیاری از مسایل بهینه‌سازی کاربردی مثل برنامه‌ریزی سیستم‌های قدرت و مسیریابی سیستم‌های ارتباط از راه دور، تعداد متغیرهای تصمیم‌گیری و محدودیت‌ها به طور معمول بسیار زیاد هستند [۹]-[۱۴]. چالش اصلی زمانی به وجود می‌آید که نیاز به یک جواب زمان حقیقی^۱ در یک مساله بهینه‌سازی با مقیاس بزرگ باشد [۱۵]. برای چنین کاربردهایی، روش‌های بهینه‌سازی کلاسیک ممکن است با توجه به افزایش بعد مساله و الگوریتم مورد استفاده در زمان محاسبات مناسب نباشد. یک راه حل بسیار امیدوار کننده، روش‌های بهینه‌سازی مبتنی بر شبکه‌های عصبی مصنوعی برای حل این نوع از مسایل است.

* عهده‌دار مکاتبات

آدرس الکترونیکی: web2_alireza.nazemi@shahroodut.ac.ir

¹Real time

شبکه‌ی عصبی از مک کلاچ و پژوهش‌های پیشگامانه‌ی نیم قرن پیش توسط پیتر [۱۶] ناشی می‌شود. از آن زمان به بعد مدل‌های شبکه عصبی مختلفی ایجاد شده است. ایده اصلی در استفاده از شبکه‌های عصبی مصنوعی برای مسایل بهینه‌سازی استفاده از یک تابع انرژی و یک سیستم دینامیکی است که این دو، بیان کننده‌ی مدل‌های شبکه عصبی مصنوعی متناظر مسایل بهینه‌سازی هستند. سیستم دینامیکی بیان شده یک دستگاه معادلات دیفرانسیل غیرخطی مرتبه اول است. انتظار می‌رود که برای یک نقطه‌ی آغازین نقطه‌ی تعادل دستگاه معادلات دیفرانسیل غیرخطی به دست آمده، جواب بهینه‌ی مساله بهینه‌سازی اصلی باشد. یک اصل اساسی در استفاده از تابع انرژی این است که برای همگرایی دستگاه معادلات دیفرانسیل به نقطه‌ی تعادل آن، تابع انرژی متناظر با آن باید نامتفق و مشتق آن نسبت به زمان باید منفی باشد. مزیت اصلی استفاده از شبکه‌های عصبی برای مسایل بهینه‌سازی این است که فرآیند یافتن جواب در یک سیستم دینامیکی به طور ذاتی به صورت موازی است؛ بنابراین روش شبکه عصبی مسایل بهینه‌سازی را به صورت در زمان^۱ حل می‌کند که خیلی سریع‌تر از الگوریتم‌های کلاسیک روی کامپیوترهای دیجیتال عمل می‌کند. مزیت دیگر روش شبکه‌های عصبی برای حل مسایل بهینه‌سازی این است که برخلاف بسیاری از روش‌های کلاسیک مانند روش سیمپلکس که وابسته به نقطه شروع هستند، این روش‌ها هیچ نوع وابستگی به نقطه شروع ندارند و به اصطلاح همگرایی سراسری هستند. در حقیقت نقطه شروع چه داخل، روی مرز یا خارج ناحیه شدنی باشد، این روش‌ها همگرا به جواب هستند.

در سال‌های اخیر، شبکه‌های عصبی زمان پیوسته‌ی بسیاری برای حل مسایل بهینه‌سازی مختلفی توسعه یافته و برخی نتایج بسیار مهم نیز به دست آمده است. به عنوان مثال می‌توان به حل مسایل برنامه‌ریزی خطی و درجه دوم، مسایل بهینه‌سازی غیرخطی، مسایل نامساوی وردشی و مسایل مکمل خطی و غیرخطی اشاره کرد [۱۷-۵۳]. در سال ۲۰۱۳، ناظمی در [۳۳] یک مدل شبکه عصبی کارا برای حل صورت کلی مسایل بهینه‌سازی غیرخطی محدب پیشنهاد کرد. از آن زمان به بعد ناظمی و همکاران [۳۴]-[۴۰] موفق شدند با حل مسایل مختلف کارایی مدل ارایه شده را نشان دهند. هدف این مقاله توسعه و اصلاح یک شبکه عصبی جدید از مدل بیان شده در [۳۳] برای حل مسایل برنامه‌ریزی غیرخطی محدب با قیود هیبریدی است. این مدل ساخته شده دارای تعداد متغیرهای کم‌تر، پیچیدگی ساختاری پایین‌تر و شرایط همگرایی ساده‌تری است. این موضوع به عنوان نوآوری اصلی این مقاله است که در بخش چهارم از مقاله به طور مفصل مورد بررسی قرار خواهد گرفت.

ساختار این مقاله به هفت بخش تقسیم می‌شود. در بخش دوم صورت کاملی از یک مساله برنامه‌ریزی غیرخطی با محدودیت‌های هیبریدی بیان می‌گردد. یک مدل شبکه عصبی کارا در بخش سوم برای حل مساله اصلی پیشنهاد می‌شود. در بخش چهارم، مقایسه کاملی بین مدل ارایه شده و برخی مدل‌های موجود بیان می‌گردد. پایداری و همگرایی سراسری مدل بیان شده به طور کامل در بخش پنجم از مقاله مورد بحث قرار می‌گیرد. در بخش ششم، نتایج شبیه‌سازی و مثال‌های به کار رفته، کارایی شبکه عصبی پیشنهادی را به طور کامل تایید می‌کنند. در نهایت مقاله با نتیجه‌گیری در بخش هفتم به پایان می‌رسد.

^۱Online

۲ بیان مساله

مساله‌ی برنامه‌ریزی غیرخطی با محدودیت‌های هیبریدی را به صورت زیر در نظر بگیرید:

$$\text{Min } f(x) \quad (1)$$

$$s.t. \quad Ax = b, \quad (2)$$

$$Hx \leq g, \quad (3)$$

$$x \in Q, \quad (4)$$

که در آن $f: \mathbb{R}^n \rightarrow \mathbb{R}$ محدب و به طور پیوسته دوبار مشتق‌پذیر است، $A \in \mathbb{R}^{l \times n}$

$$\text{rank}(A) = l \text{ و } g \in \mathbb{R}^m$$

$$Q = \left\{ x \in \mathbb{R}^n : \alpha_i \leq x_i \leq \beta_i \quad (i=1, \dots, n) \right\},$$

که در آن β_i ها می‌توانند $+\infty$ و α_i ها می‌توانند $-\infty$ باشند. فرض کنید

$$D = \text{diag}(1, 1, \dots, 1) \in \mathbb{R}^{n \times n},$$

$$B = \begin{pmatrix} H \\ D \\ -D \end{pmatrix} \in \mathbb{R}^{(m+n) \times n},$$

$$\beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} \in \mathbb{R}^n,$$

$$\alpha = \begin{pmatrix} -\alpha_1 \\ \vdots \\ -\alpha_n \end{pmatrix} \in \mathbb{R}^n,$$

$$d = \begin{pmatrix} g \\ \beta \\ \alpha \end{pmatrix} \in \mathbb{R}^{m+2n}.$$

در این صورت مساله (۱)-(۴) به صورت زیر تبدیل می‌شود:

$$\text{Min } f(x) \quad (5)$$

$$s.t. \quad Ax = b, \quad (6)$$

$$Bx \leq d. \quad (7)$$

قضیه ۱-۲ [۳] $x^* \in \mathbb{R}^n$ یک جواب بهینه (۵)-(۷) است اگر و فقط اگر $v^* \in \mathbb{R}^{m+n}$ و $u^* \in \mathbb{R}^l$ وجود داشته

باشد به طوری که $(x^{*T}, u^{*T}, v^{*T})^T$ در شرایط کاروش-کان-تاکر^۱ (KKT) زیر صدق کند:

¹Karush-Kuhn-Tucker

$$\begin{cases} u^* \geq 0, & Bx^* - d \leq 0, & u^{*T} (Bx^* - d) = 0, \\ \nabla f(x^*) + B^T u^* + A^T v^* = 0, \\ Ax^* = b. \end{cases} \quad (8)$$

x^* یک نقطه KKT مساله (۵)–(۷) نامیده می‌شود و زوج $(u^{*T}, v^{*T})^T$ بردار ضرایب لگرانژ متناظر با x^* نامیده می‌شود.

قضیه ۲-۲ x^* یک جواب بهینه (۵)–(۷) است اگر و فقط اگر x^* یک نقطه KKT از (۵)–(۷) باشد. [۳]

۳ مدل شبکه عصبی

اگر چه انواع مختلفی از شبکه‌های عصبی برای محاسبه وجود دارند، اما تاکنون فقط دو مدل برای حل مسایل بهینه‌سازی محدب با قیود هیبریدی ارایه شده است [۴۹، ۴۱]. با عنایت به اینکه ساختار این دو مدل نیاز به l متغیر وضعیت برای برقراری قیود تساوی در شرایط بهینگی (۸) دارد، ساختار این دو مدل بیان شده پیچیده است و نیاز به ساده‌سازی این ساختارها وجود دارد؛ بنابراین هدف ما حذف این l متغیر وضعیت و ساختن یک مدل شبکه عصبی با ساختاری ساده‌تر، پایداری تضمین شده و همگرایی سراسری است. برای ساده‌سازی ساختار شبکه ابتدا لم زیر را در نظر می‌گیریم.

لم ۱-۳ x^* جواب بهینه‌ی (۵)–(۷) است اگر و تنها اگر $u^* \geq 0$ موجود باشد به‌طوری که $(x^{*T}, u^{*T})^T$ در شرایط زیر صدق کند:

$$(I - P) [\nabla f(x^*) + B^T u^*] + Q(Ax^* - b) = 0, \quad (9)$$

$$(u^* + Bx^* - d)^+ - u^* = 0. \quad (10)$$

برهان: اگر x^* جواب بهینه‌ی (۵)–(۷) باشد، مطابق شرایط بهینگی (۸) برای مسایل بهینه‌سازی محدب، u^* موجودند به‌طوری که در دستگاه زیر صدق کنند:

$$\nabla f(x^*) + A^T v^* + B^T u^* = 0, \quad (11)$$

$$Ax^* = b, \quad (12)$$

$$d - Bx^* \geq 0, \quad u^* \geq 0, \quad (u^*)^T (d - Bx^*) = 0. \quad (13)$$

به آسانی دیده می‌شود که (۱۰) و (۱۳) معادل هستند. به این معنی که:

$$d - Bx^* \geq 0, \quad u^* \geq 0, \quad (u^*)^T (d - Bx^*) = 0 \Leftrightarrow (u^* + Bx^* - d)^+ - u^* = 0.$$

در ادامه، نشان می‌دهیم که جواب‌های (۹)، (۱۱) و (۱۲) معادلند. از (۱۱) داریم:

$$-A(\nabla f(x^*) + A^T v^* + B^T u^*) = 0.$$

با جمع کردن معادله‌ی بالا با (۱۲) داریم:

$$Ax^* - A(\nabla f(x^*)) - AA^T v^* - AB^T u^* = b.$$

بنابراین:

$$A^T v^* = A^T (AA^T)^{-1} (Ax^* - b) - A^T (AA^T)^{-1} (\nabla f(x^*) + B^T u^*). \quad (14)$$

با جایگذاری (11) در (14) داریم:

$$[I - A^T (AA^T)^{-1} A] (\nabla f(x^*) + B^T u^*) + A^T (AA^T)^{-1} (Ax^* - b) = 0. \quad (15)$$

فرض کنید A می‌توان (15) را به صورت زیر نوشت:

$$(I - P) [\nabla f(x^*) + B^T u^*] + Q (Ax^* - b) = 0.$$

بر عکس، اگر u^* موجود باشد به طوری که $(x^{*T}, u^{*T})^T$ در (9) صدق کند با ضرب A در طرفین (9) داریم:

$$A(I - P) [\nabla f(x^*) + B^T u^*] + AQ (Ax^* - b) = 0.$$

توجه کنید که $AQ = I$ و $A(I - P) = 0$ داریم:

$$(I - P) (\nabla f(x^*) + B^T u^*) = 0,$$

$$Ax^* - b = 0.$$

فرض کنید $v^* = -(AA^T)^{-1} A (\nabla f(x^*) + B^T u^*)$ ؛ بنابراین داریم:

$$\nabla f(x^*) + A^T v^* + B^T u^* = \nabla f(x^*) + B^T u^* - A^T (AA^T)^{-1} A (\nabla f(x^*) + B^T u^*)$$

$$= (I - P) (\nabla f(x^*) + B^T u^*) = 0.$$

بنابراین $(x^{*T}, u^{*T}, v^{*T})^T$ وجود دارد به طوری که (11) و (12) در شرایط بهینگی (8) صدق می‌کند. پس حکم ثابت شده است.

می‌توان با استفاده از لم ۳-۱ مدل شبکه عصبی زیر را برای به دست آوردن جواب بهینه مساله (5)-(7) به صورت زیر به کار برد:

$$\begin{cases} \frac{dx}{dt} = -(I - P)[\nabla f(x) + B^T(u + Bx - d)^+] - Q(Ax - b), \\ \frac{du}{dt} = -u + (u + Bx - d)^+, \end{cases} \quad (16)$$

که در آن:

$$P = A^T (AA^T)^{-1} A, \quad Q = A^T (AA^T)^{-1},$$

$$(u + Bx - d)^+ = ([u + Bx - d]_1^+, [u + Bx - d]_2^+, \dots, [u + Bx - d]_{m+n}^+),$$

$$[u + Bx - d]_k^+ = \max\{(u + Bx - d)_k, 0\}, \quad k = 1, 2, \dots, m+n.$$

توجه ۳-۲ با استفاده از لم ۳-۱، به آسانی می‌بینیم که x^* جواب بهینه مساله (5)-(7) است اگر و تنها اگر u^* وجود داشته باشد به طوری که $(x^{*T}, u^{*T})^T$ نقطه تعادل شبکه عصبی ارایه شده در (16) باشد؛ بنابراین، زمانی که شبکه عصبی به نقطه تعادل همگراست، مسیر حرکت (t) به جواب بهینه مساله (5)-(7) همگرا می‌باشد.

۴ مقایسه با برخی مدل‌های موجود

برای اینکه بینیم مدل شبکه‌ی عصبی ارایه شده در (۱۶) چه مزیت‌هایی در حل مساله بهینه‌سازی (۵)–(۷) دارد، آن را با برخی مدل‌های موجود مقایسه می‌کنیم.

با استفاده از مدل ارایه شده در [۲۳] از مدل شبکه‌ی عصبی زیر می‌توان برای حل (۵)–(۷) استفاده کرد:

$$\frac{dx}{dt} = -\left(x - P_{\Omega}\left(x - \nabla f(x) - B^T u + A^T v\right)\right), \quad (17)$$

$$\frac{du}{dt} = -(u - (u + Bx - d)^+), \quad (18)$$

$$\frac{dv}{dt} = Ax - b, \quad (19)$$

که در آن $\mathbb{R}^n \subset \Omega$ یک مجموعه محدب بسته و $P_{\Omega}: \mathbb{R}^n \rightarrow \Omega$ عملگر تصویر است [۲۴] که به صورت

زیر تعریف می‌شود:

$$P_{\Omega} = \operatorname{argmin}_{v \in \Omega} \|x - v\|.$$

توجه کنید که هیچ تضمینی برای پایداری مدل دینامیکی (۱۷)–(۱۹) برای زمانی که f یک تابع محدب است وجود ندارد [۲۴]; بنابراین دستگاه دینامیکی مطرح شده در (۱۷)–(۱۹) نمی‌تواند یک مساله برنامه‌ریزی خطی را حل کند. برای توضیحات بیشتر، مثال ۱–۴ در [۳۳] را ببینید.

نوع دیگری از مدل شبکه عصبی، مدل گرادیانی است. در مدل شبکه عصبی گرادیانی، یک مساله بهینه‌سازی مقید را می‌توان با یک مساله بهینه‌سازی نامقید تقریب زد. سپس می‌توان تابع انرژی را با استفاده از روش تابع جریمه ساخت. لازم به ذکر است که مدل شبکه عصبی گرادیانی نسبت به مدل‌هایی که با استفاده از تعریف مستقیم مشتقات تابع انرژی بیان شده‌اند دارای مزیت‌هایی است؛ اما نقطه ضعف آن در عدم تضمین همگرایی است [۵۲]، چون این مدل‌ها که بر مبنای تابع جریمه ساخته می‌شوند، نیاز به یک پارامتر قابل تنظیم^۱ به نام پارامتر جریمه دارند. تقریب مساله مقید (۵)–(۷) با استفاده از مساله بهینه‌سازی نامقید زیر، یک نمونه از کاربرد روش تابع جریمه است:

$$\text{Min } E_1(x) = f(x) + \frac{\gamma}{2} \left\{ \sum_{k=1}^{m+n} [(Bx - d)_k^+]^\gamma + \sum_{p=1}^l (Ax - b)_p^\gamma \right\},$$

که در آن γ پارامتر جریمه است. مدل شبکه عصبی گرادیانی با روش زیر به دست می‌آید:

$$\frac{dx}{dt} = -\nabla E_1(x) = -(\nabla f(x) + \gamma \left[B^T (Bx - d)^+ + A^T (Ax - b) \right]). \quad (20)$$

سیستم ارایه شده در (۲۰) به مدل کندی و چاو^۲ معروف است که در [۱۸] بیان شده است. این شبکه عصبی به علت دارا بودن پارامتر جریمه‌ای متناهی، توانایی یافتن یک جواب بهینه دقیق را ندارد و اگر پارامتر جریمه‌ای خیلی بزرگ باشد، یافتن یک جواب بهینه دقیق دشوار است [۲۸]. همچنین می‌توان نشان داد که شبکه عصبی

¹Adjustable

²Kennedy and Chua

(۲۰) در بعضی مسائل برنامه‌ریزی محدب، همگرایی سراسری به یک جواب بهینه دقیق نیست. برای توضیح بیشتر مثال ۱-۵ در [۳۰] را بینید.

با شرط محدب اکید بودن $f(x)$ ، مدل شبکه عصبی گرادیانی زیر در [۳۱] برای حل (۵)-(۷) ارایه می‌شود:

$$\frac{dx}{dt} = -\left(\nabla f(x) + \frac{1}{\gamma} B^T u + A^T v \right), \quad (21)$$

$$\frac{du}{dt} = \text{diag}(u_1, \dots, u_{m+n})(Bx - d), \quad (22)$$

$$\frac{dv}{dt} = Ax - b, \quad (23)$$

با نقطه اولیه $(x_0^T, u_0^T, v_0^T)^T$. پایداری بر حسب لیاضانف و همگرایی سراسری این مدل با شرط محدب اکید بودن $f(x)$ تضمین می‌شود؛ بنابراین مدل نمی‌تواند مسائل برنامه‌ریزی خطی را حل کند.

می‌توانیم بین جواب مساله (۵)-(۷) و جواب یک مساله مینیمم‌سازی نامقید همارز توسط یک تابع شایسته^۱ ارتباط ایجاد کنیم.

تعریف ۴-۱ تابع $\phi: \mathbb{R}^2 \rightarrow \mathbb{R}$ یک تابع NCP نامیده می‌شود اگر در شرایط زیر صدق کند [۳۹]:

$$\phi(a, b) = 0 \Leftrightarrow a \geq 0, b \geq 0, ab = 0.$$

یک تابع NCP متداول، تابع فیشربرمیستر^۲ است که قویاً نیمه هموار است و به صورت زیر تعریف می‌شود:

$$\phi_{FB}^\varepsilon(a, b) = (a + b) - \sqrt{a^2 + b^2 + \varepsilon}.$$

تعریف ۴-۲ تابع شایسته تابعی است که مینیمم سراسری آن منطبق بر جواب‌های NCP است. [۳۹]

تابع شایسته معروف $\psi_{FB}: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}_+$ که به وسیله مجذور ϕ_{FB}^ε به دست می‌آید، به صورت زیر است:

$$\psi_{FB}(a, b) = \frac{1}{2} \|\phi_{FB}^\varepsilon(a, b)\|^2.$$

اکنون با استفاده از [۳۹] به راحتی درمی‌یابیم که جواب مساله (۵)-(۷) معادل نقطه تعادل شبکه عصبی زیر است:

$$\frac{d(x(t), u(t), v(t))}{dt} = -\nabla E(x(t), u(t), v(t)), \quad (24)$$

که در آن:

$$E(x, u, v) = \frac{1}{2} \|\eta(x, u, v)\|^2,$$

$$\eta(x, u, v) = \begin{bmatrix} \nabla f(x) + A^T v + B^T u \\ b - Ax \\ \phi_{FB}^\varepsilon(u, d - Bx) \end{bmatrix}.$$

¹Merit function

²Fisher Burmister

پایداری و همگرایی سراسری این مدل نیز با شرط محدب اکید بودن $f(x)$ تضمین می‌شود.

در [۳۳] یک مدل شبکه عصبی کارا برای حل مساله (۵)–(۷) به صورت زیر بیان شده است:

$$\frac{d(x(t), u(t), v(t))}{dt} = \phi(x(t), u(t), v(t)), \quad (25)$$

که در آن:

$$\phi(x, u, v) = \begin{cases} -(\nabla f(x) + B^T(u + Bx - d)^+ + A^T v) \\ (u + Bx - d)^+ - u \\ Ax - b \end{cases}. \quad (26)$$

در این مدل $v \in \mathbb{R}^l$ و $x \in \mathbb{R}^n, u \in \mathbb{R}^{m+n}$ است؛ بنابراین شبکه عصبی (۲۵) دارای $m+3n+l$ نرون است؛ اما مدل شبکه عصبی (۱۶) دارای $m+3n$ نرون است. در حقیقت مدل (۱۶) یک اصلاح از مدل (۲۵) با تعداد نرون کمتر و ساختار ساده‌تر است.

۵ تحلیل پایداری

در این بخش نشان می‌دهیم که مدل شبکه‌ی عصبی (۱۶) همگرای سراسری است.

لم ۱-۵ فرض کنید $(x^{*T}, u^{*T})^T \in \Omega^e$ یک نقطه‌ی تعادل شبکه‌ی عصبی (۱۶) باشد و

$$V(x, u) = f(x) - f(x^*) + \frac{1}{2} \| (u + Bx - d)^+ \|_r^r - \frac{1}{2} \| (u^* + Bx^* - d)^+ \|_r^r - (u - u^*)^T u^* - (x - x^*)^T (\nabla f(x^*) + B^T u^*) + \frac{1}{2} \| x - x^* \|_r^r + \frac{1}{2} \| u - u^* \|_r^r. \quad (27)$$

سپس

الف-

$$V(x, u) \geq \frac{1}{2} \| x - x^* \|_r^r + \frac{1}{2} \| u - u^* \|_r^r; \quad (28)$$

ب-

$$\begin{aligned} \frac{dV}{dt} &\leq - \| (I - P)[\nabla f(x) + B^T(u + Bx - d)^+] + Q(Ax - b) \|_r^r \\ &\quad - \frac{1}{2} \| u - (u + Bx - d)^+ \|_r^r. \end{aligned} \quad (29)$$

برهان: الف- چون تابع $f(x) + \frac{1}{2} \| (u + Bx - d)^+ \|_r^r$ محدب و مشتق‌پذیر است و

$$\nabla(f(x) + \frac{1}{2} \| (u + Bx - d)^+ \|_r^r) = \begin{bmatrix} \nabla f(x) + B^T(u + Bx - d)^+ \\ (u + Bx - d)^+ \end{bmatrix},$$

در نتیجه:

$$f(x) - f(x^*) + \frac{1}{2} \| (u + Bx - d)^+ \|_r^r - \frac{1}{2} \| (u^* + Bx^* - d)^+ \|_r^r$$

$$\geq (x - x^*)^T \left(\nabla f(x^*) + B^T u^* \right) + (u - u^*)^T u^*.$$

بنابراین:

$$V(x, u) \geq \frac{1}{\gamma} \|x - x^*\|^\gamma + \frac{1}{\gamma} \|u - u^*\|^\gamma.$$

ب- برای اثبات (۲۹) داریم:

$$\begin{aligned} \frac{dV}{dt} &= \frac{\partial V}{\partial x} \frac{dx}{dt} + \frac{\partial V}{\partial u} \frac{du}{dt} \\ &= -[\nabla f(x) + B^T(u^* + Bx^* - d)^+ - \nabla f(x^*) - B^T u^* + x - x^*]^T \\ &\quad \times \{(I - P)[\nabla f(x) + B^T(u + Bx - d)^+] + Q(Ax - b)\} \\ &\quad - \frac{1}{\gamma}[u - (u + Bx - d)^+]^T [(u + Bx - d)^+ - \gamma u^* + u] \\ &= -[\nabla f(x) - \nabla f(x^*) + x - x^* + B^T(u + Bx - d)^+ - B^T u^*]^T \\ &\quad \times \{(I - P)[\nabla f(x) - \nabla f(x^*) + B^T(u + Bx - d)^+ - B^T u^*] + P(x - x^*)\} \\ &\quad - \frac{1}{\gamma}[u - (u + Bx - d)^+]^T [u - (u + Bx - d)^+ + \gamma(u + Bx - d)^+ - \gamma u^*] \\ &= -\{[\nabla f(x) - \nabla f(x^*)]^T (I - P)[\nabla f(x) - \nabla f(x^*)] \\ &\quad + [\nabla f(x) - \nabla f(x^*)]^T (I - P)[B^T(u + Bx - d)^+ - B^T u^*] \\ &\quad + [\nabla f(x) - \nabla f(x^*)]^T P(x - x^*) \\ &\quad + (x - x^*)^T (I - P)[\nabla f(x) - \nabla f(x^*)] \\ &\quad + (x - x^*)^T (I - P)[B^T(u + Bx - d)^+ - B^T u^*] + (x - x^*)^T P(x - x^*) \\ &\quad + [B^T(u + Bx - d)^+ - B^T u^*]^T (I - P)[\nabla f(x) - \nabla f(x^*)] \\ &\quad + [B^T(u + Bx - d)^+ - B^T u^*]^T (I - P)[B^T(u + Bx - d)^+ - B^T u^*] \\ &\quad + [B^T(u + Bx - d)^+ - B^T u^*]^T P(x - x^*)\} \\ &\quad - \frac{1}{\gamma} \|u - (u + Bx - d)^+\|^\gamma + [(u + Bx - d)^+ - u]^T [(u + Bx - d)^+ - u^*]. \end{aligned}$$

توجه کنید که:

$$(I - P)^\gamma = I - P, P^\gamma = P, P(I - P) = 0,$$

و

$$(u + Bx - d)^+ - u = Bx - d + (-u - (Bx - d))^+.$$

بنابراین داریم:

$$\begin{aligned} \frac{dV}{dt} &= -[\nabla f(x) - \nabla f(x^*)]^T (I - P)^\gamma [\nabla f(x) - \nabla f(x^*)] \\ &\quad - \gamma [\nabla f(x) - \nabla f(x^*)] (I - P)^\gamma [B^T(u + Bx - d)^+ - B^T u^*] \end{aligned}$$

$$\begin{aligned}
 & -[B^T(u+Bx-d)^+ - B^Tu^*]^T(I-P)^r[B^T(u+Bx-d)^+ - B^Tu^*] \\
 & -(x-x^*)^T P^r(x-x^*) - (x-x^*)^T [\nabla f(x) - \nabla f(x^*)] \\
 & -(x-x^*)^T [B^T(u+Bx-d)^+ - B^Tu^*] \\
 & -\frac{1}{r} \|u - (u+Bx-d)^+\|^r + [Bx-d + (-u-Bx+d)^+]^T [(u+Bx-d)^+ - u^*] \\
 = & -\|(I-P)[\nabla f(x) - \nabla f(x^*)]\| + (I-P)[B^T(u+Bx-d)^+ - B^Tu^*] + P(x-x^*) \\
 & -(x-x^*)^T [\nabla f(x) - \nabla f(x^*)] - (x-x^*)^T [B^T(u+Bx-d)^+ - B^Tu^*] \\
 & -\frac{1}{r} \|u - (u+Bx-d)^+\|^r + (Bx-d)^T(u+Bx-d)^+ - (Bx-d)^T u^* \\
 & + [(-u-(Bx-d))^+]^T(u+Bx-d)^+ - [(-u-(Bx-d))^+]^T u^* \\
 = & -\|(I-P)[\nabla f(x) + B^T(u+Bx-d)^+]\| + Q(Ax-b) \\
 & -(x-x^*)^T [\nabla f(x) - \nabla f(x^*)] + (x-x^*)^T B^Tu^* - (x-x^*)^T B^T(u+Bx-d)^+ \\
 & -\frac{1}{r} \|u - (u+Bx-d)^+\|^r + (Bx-d)^T[(u+Bx-d)^+] - (Bx-d)^T u^* \\
 & + [(-u-(Bx-d))^+]^T(u+Bx-d)^+ - [(-u-(Bx-d))^+]^T u^* \\
 = & -\|(I-P)[\nabla f(x) + B^T(u+Bx-d)^+]\| + Q(Ax-b) \\
 & -\frac{1}{r} \|u - (u+Bx-d)^+\|^r - (x-x^*)^T [\nabla f(x) - \nabla f(x^*)] - [(u+Bx-d)^+]^T u^* \\
 & -(Bx^*-d)^T u^* + (Bx^*-d)^T(u+Bx-d)^+.
 \end{aligned}$$

چون تابع $f(x)$ محدب است؛ بنابراین داریم:

$$(x-x^*)^T [\nabla f(x) - \nabla f(x^*)] \geq 0.$$

به علاوه به آسانی می‌توان تحقیق کرد که:

$$[(-u-(Bx-d))^+]^T(u+Bx-d)^+ = 0,$$

$$(Bx^*-d)^T u^* = 0,$$

$$[(-u-(Bx-d))^+]^T u^* \geq 0,$$

$$g(x^*)^T [(u+Bx-d)^+]^T \leq 0.$$

بنابراین:

$$\frac{dV}{dt} \leq -\|(I-P)[\nabla f(x) + B^T(u+Bx-d)^+]\| + Q(Ax-b) - \frac{1}{r} \|u - (u+Bx-d)^+\|^r \leq 0.$$

توجه ۲-۵ معرفی شده در (۲۷) یک تابع لیپانوف است، بنابراین، شبکه‌ی عصبی (۱۶) پایدار به مفهوم لیپانوف است.

قضیه ۳-۵ برای هر نقطه‌ی اولیه‌ی $x(\cdot)^T, u(\cdot)^T \in \mathbb{R}^{m+n}$ ، یک جواب پیوسته منحصر به فرد $x(t)^T, u(t)^T \in \mathbb{R}^{m+n}$ وجود دارد.

برهان: چون $f(x)$ روی \mathbb{R}^n به طور پیوسته مشتق‌پذیر است، $(\nabla f(x))$ پیوسته لیپ شیتر محلی است. درنتیجه $(I-P)[\nabla f(x) + B^T(u+Bx-d)^+ - u]$ پیوسته لیپ شیتر محلی هستند. از قضیه‌ی وجودی جواب موضعی برای معادلات دیفرانسیل معمولی [۵۵] نتیجه می‌شود یک جواب منحصر به فرد از (۱۶) روی $[t_0, T]$ وجود دارد. بنا به لم ۵-۱ می‌دانیم که V نسبت به t یک تابع نزولی است؛ لذا:

$$\frac{1}{2} \|x - x^*\|^r + \frac{1}{2} \|u - u^*\|^r \leq V(x(\cdot), u(\cdot)) \leq V(x(t), u(t)), \quad \forall t \geq 0, \quad (30)$$

نشان می‌دهد مسیر حالت شبکه‌ی عصبی $x(t)^T, u(t)^T$ از (۱۶) روی $[t_0, T]$ کراندار است. درنتیجه $T = +\infty$ این اثبات را کامل می‌کند.

قضیه ۴-۴ مسیر حالت شبکه‌ی عصبی (۱۶) برای هر نقطه‌ی اولیه‌ی $x(\cdot)^T, u(\cdot)^T \in \mathbb{R}^{m+n}$ به یک نقطه‌ی تعادل همگرا است. به ویژه، شبکه‌ی عصبی (۱۶) با هر نقطه‌ی اولیه‌ی $x(\cdot)^T, u(\cdot)^T \in \mathbb{R}^{m+n}$ هنگامی که نقطه‌ی تعادل منحصر به فرد دارد پایدار مجانبی سراسری است.

برهان: تعریف کنید:

$$D(x, u) = \| (I - P)[\nabla f(x) + B^T(u + Bx - d)^+] + Q(Ax - b) \|^r \\ + \frac{1}{2} \|u - (u + Bx - d)^+\|^r.$$

در این صورت $D(x, u) = 0$ اگر و تنها اگر $(x^T, u^T)^T$ یک نقطه‌ی تعادل شبکه‌ی عصبی (۱۶) باشد. از اثبات قضیه‌ی ۳-۵ نتیجه می‌گیریم مسیر حالت $(x(t)^T, u(t)^T)$ شبکه‌ی عصبی (۱۶) کراندار است؛ بنابراین، دنباله‌ی صعودی $\{t_n\}$ با $\lim_{n \rightarrow \infty} t_n \rightarrow \infty$ و نقطه‌ی حدی $(\hat{x}^T, \hat{u}^T)^T$ وجود دارند به‌طوری که $\hat{x} \rightarrow \hat{x}(t_n)$ و $\hat{u} \rightarrow \hat{u}(t_n)$. حال ثابت می‌کنیم $(\hat{x}^T, \hat{u}^T)^T \in \Omega^e$. مسیر حالت $(\hat{x}^T, \hat{u}^T)^T$ همگرای سراسری به نقطه‌ی تعادل $(\hat{x}^T, \hat{u}^T)^T$ است. در مرحله اول، ثابت می‌کنیم که $D(\hat{x}, \hat{u}) = 0$ ؛ یعنی $D(\hat{x}, \hat{u}) = 0$ یک نقطه‌ی تعادل شبکه‌ی عصبی (۱۶) است. فرض کنید چنین نباشد (برهان خلف)؛ یعنی $D(\hat{x}, \hat{u}) > 0$ چون $D(x, u) = 0$ به ترتیب نسبت به x و u پیوسته است، $\varepsilon > 0$ و ε -همسايگی:

$$B((\hat{x}, \hat{u}), \varepsilon) = \{(x^T, u^T)^T : \|x - \hat{x}\| + \|u - \hat{u}\| \leq \varepsilon\},$$

از $D(x, u) > q$ و وجود دارند به‌طوری که برای هر $(x^T, u^T)^T \in B((\hat{x}, \hat{u}), \varepsilon)$ از طرفی با توجه به اینکه $\hat{x} \rightarrow \hat{x}(t_n)$ و $\hat{u} \rightarrow \hat{u}(t_n)$ عدد صحیح مثبت N وجود دارد به‌طوری که برای هر $n \geq N$ نتیجه $\|u(t_n) - \hat{u}\| \leq \frac{1}{4}\varepsilon$ و $\|x(t_n) - \hat{x}\| \leq \frac{1}{4}\varepsilon$ باشد. از (۱۶) و کراندار بودن $(x(t)^T, u(t)^T)^T$ نتیجه می‌گیریم \hat{x} و \hat{u} نیز کراندار هستند که این کران با M نشان داده می‌شود. با در نظر گرفتن:

$$t \in \left[t_n - \frac{\varepsilon}{\lambda M}, t_n + \frac{\varepsilon}{\lambda M} \right], n \geq N,$$

داریم:

$$\begin{aligned} & \|x(t) - \hat{x}\| + \|u(t) - \hat{u}\| \\ & \leq \|x(t) - x(t_n)\| + \|u(t) - u(t_n)\| + \|x(t_n) - \hat{x}\| + \|u(t_n) - \hat{u}\| \\ & = \|\dot{x}(\xi_1)\| \times |t - t_n| + \|\dot{u}(\xi_2)\| \times |t - t_n| + \|x(t_n) - \hat{x}\| + \|u(t_n) - \hat{u}\| \\ & \leq 2M |t - t_n| + \frac{\varepsilon}{\gamma} \leq \varepsilon. \end{aligned}$$

از این رو، برای هر $(x^T, u^T)^T \in B((\hat{x}, \hat{u}), \varepsilon)$ ؛ بنابراین برای $n \geq N$ ، $t \in \left[t_n - \frac{\varepsilon}{\lambda M}, t_n + \frac{\varepsilon}{\lambda M} \right]$

و $\lim_{n \rightarrow \infty} t_n = \infty$ داریم. چون $D(x(t), u(t)) > q$ نتیجه می‌شود $n \geq N$ ، $t \in \left[t_n - \frac{\varepsilon}{\lambda M}, t_n + \frac{\varepsilon}{\lambda M} \right]$ هر

اندازه‌ی لبگ^۱ مجموعه‌ی $t \in \cup_{n \geq N} \left[t_n - \frac{\varepsilon}{\lambda M}, t_n + \frac{\varepsilon}{\lambda M} \right]$ داریم:

$$\int D(x(t), u(t)) dt = \infty. \quad (31)$$

از طرفی:

$$\begin{aligned} \int D(x(t), u(t)) dt &= \lim_{s \rightarrow \infty} \int D(x(t), u(t)) dt \leq - \lim_{s \rightarrow \infty} \int \dot{V}(x(t), u(t)) dt \\ &= \lim_{s \rightarrow \infty} [V(x(\cdot), u(\cdot)) - V(x(s), u(s))] \leq V(x(\cdot), y(\cdot)), \end{aligned}$$

که (31) را نقض می‌کند؛ لذا $D(\hat{x}, \hat{u}) = 0$. در مرحله دوم، ثابت می‌کنیم مسیر حالت $(x(t)^T, u(t)^T)$ همگرای سراسری به نقطه‌ی تعادل (\hat{x}^T, \hat{u}^T) است. تابع لیپانوف زیر را تعریف کنید:

$$\begin{aligned} \hat{V}(x, u) &= f(x) - f(\hat{x}) + \frac{1}{\gamma} \| (u + Bx - d)^+ \|^r - \frac{1}{\gamma} \| (\hat{u} + B\hat{x} - d)^+ \|^r - (u - \hat{u})^T \hat{u} \\ &\quad - (x - \hat{x})^T (\nabla f(\hat{x}) + B^T \hat{u}) + \frac{1}{\gamma} \| x - \hat{x} \|^r + \frac{1}{\gamma} \| u - \hat{u} \|^r. \end{aligned}$$

داریم $\hat{V}(\hat{x}, \hat{u}) = 0$. چون $\lim_{n \rightarrow \infty} u(t_n) \rightarrow \hat{u}$ و $\lim_{n \rightarrow \infty} x(t_n) \rightarrow \hat{x}$ دارد به

طوری که $\varepsilon > 0$ داریم $\hat{V}(x(t_k), u(t_k)) < \varepsilon$ بنابراین

$$\frac{1}{\gamma} \|x(t) - \hat{x}\|^r + \frac{1}{\gamma} \|u(t) - \hat{u}\|^r \leq \hat{V}(x(t), u(t)),$$

و \hat{V} نزولی است؛ بنابراین، برای هر $t \geq t_k$ داریم:

$$\frac{1}{\gamma} \|x(t) - \hat{x}\|^r + \frac{1}{\gamma} \|u(t) - \hat{u}\|^r \leq \hat{V}(x(t), u(t)) \leq \hat{V}(x(t_k), u(t_k)) \leq \varepsilon,$$

¹Lebesgue measure

یعنی $\lim_{n \rightarrow \infty} u(t) = \hat{y}$ و $\lim_{n \rightarrow \infty} x(t) = \hat{x}$ ؛ لذا مسیر حالت شبکه عصبی (۱۶) همگرای سراسری به نقطهٔ تعادل $(\hat{x}^T, \hat{u}^T)^T$ است. به ویژه، اگر $\Omega^e = \{(x^{*T}, u^{*T})^T\}$ ، آنگاه بنا به تحلیل فوق مسیر حالت $(x(t)^T, u(t)^T)^T$ با هر نقطهٔ اولیهٔ $(x^{*T}, u^{*T})^T$ میل خواهد کرد. پس این شبکه عصبی پایدار مجانبی سراسری است.

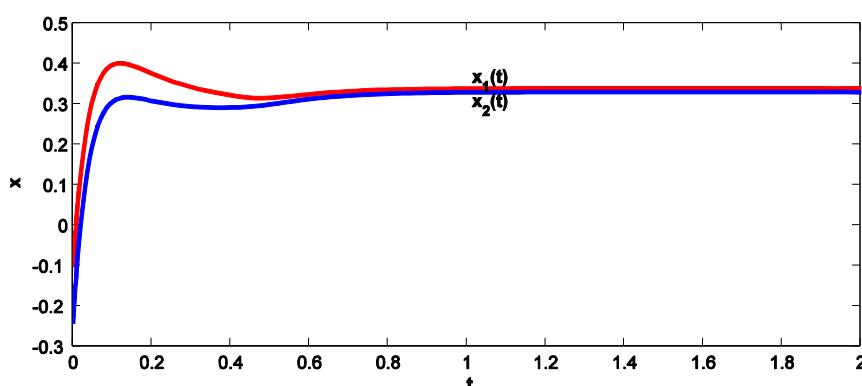
۶ شبیه‌سازی‌های عددی

برای اثبات کارایی و موثر بودن شبکه عصبی پیشنهادی، در این بخش، چندین مثال را توسط شبکه عصبی (۱۶) آزمایش می‌کنیم. شبیه‌سازی توسط نرم افزار Matlab 7 اجرا می‌شود. حل کننده معادله دیفرانسیل معمولی ode45 است. دقت جواب مثال‌های زیر توسط نرم افزار Lingo 11 احراز شده است.

مثال ۶

$$\begin{aligned} \text{Min } f(x) &= 0.4x_1 + x_1^2 + x_2^2 - x_1 x_2 + \frac{1}{3}x_1^3 \\ \text{s.t. } &\begin{cases} x_1 \geq 0, \\ x_2 \geq 0, \\ x_1 + 0.5x_2 \geq 0.4, \\ 0.5x_1 + x_2 \geq 0.5. \end{cases} \end{aligned}$$

جواب بهینه $x^* = (0.3395, 0.3303)^T$ است. شبکه عصبی (۱۶) برای حل این مساله برنامه ریزی ریاضی به کار برده می‌شود. نتایج شبیه‌سازی نشان می‌دهد که مسیرهای شبکه عصبی (۱۶) همگرا به $y^* = (x^{*T}, u^{*T})^T$ هستند. شکل ۱ نشان می‌دهد که مسیرها همگرا به جواب بهینه x^* هستند.

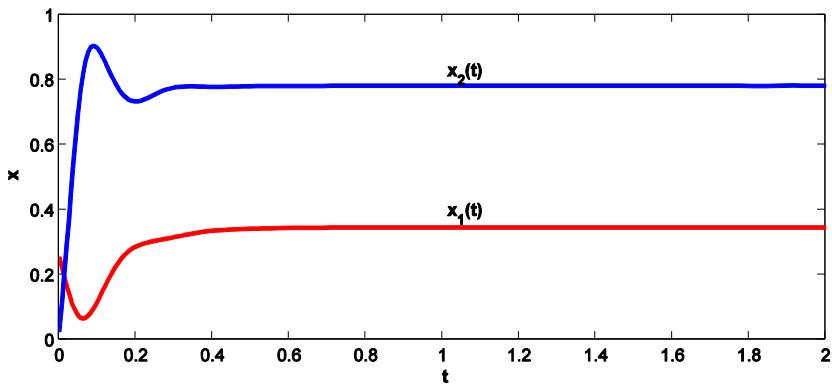


شکل ۱. رفتارهای گذراي x_1 و x_2 از شبکه عصبی (۱۶) در مثال (۶).

مثال ۲-۶

$$\begin{aligned} \text{Min } f(x) &= \frac{1}{4}x_1^4 + \dots / 5x_1^2 + \frac{1}{4}x_2^4 + \dots / 5x_2^2 - \dots / 9x_1x_2 \\ \text{s.t. } &\begin{cases} -x_1 - x_2 \geq -2, \\ x_1 - x_2 \geq -2, \\ x_1 - 3x_2 = -2, \\ 0 \leq x \leq 1. \end{cases} \end{aligned}$$

نتایج شبیه‌سازی نشان می‌دهد که مسیرهای شبکه عصبی (۱۶) همگرا به جواب بهینه $x^* = (0/346, 0/7822)^T$ است. به عنوان مثال، شکل ۲ نشان می‌دهد که مسیرهای شبکه عصبی همگرا به جواب بهینه مساله هستند.



شکل ۲. رفتارهای گذراي x_1 و x_2 از شبکه عصبی (۱۶) در مثال (۲-۶).

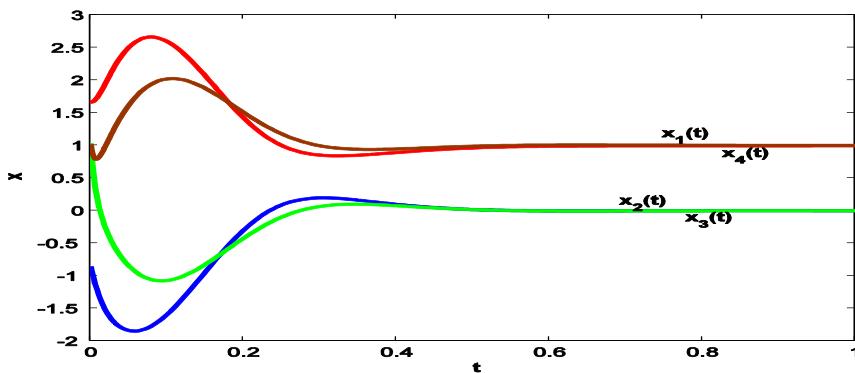
مثال ۳-۶

$$\begin{aligned} \text{Min } f(x) &= \frac{1}{2}(x_1^2 + x_2^2) + 2(x_1^2 + x_3^2) - \ln(x_1x_2) + 3x_1x_2 + 4x_2x_3 - 2x_1 - 3x_3 \\ \text{s.t. } &\begin{cases} x_1 + x_2 = 1, \\ x_2 + x_3 = 1, \\ x \in Q, \end{cases} \end{aligned}$$

که در آن:

$$Q = \{x \in \mathbb{R}^4 \mid 0 \leq x_i \leq 1, \quad i = 1, 2, 3, 4\}.$$

این مساله فقط یک جواب $x^* = (1, 0, 0, 0)^T$ دارد. شکل ۳ مسیرهای وضعیت $(x_1(t), x_2(t), x_3(t), x_4(t))^T$ را نشان می‌دهد. این بردار به x^* همگرا است.

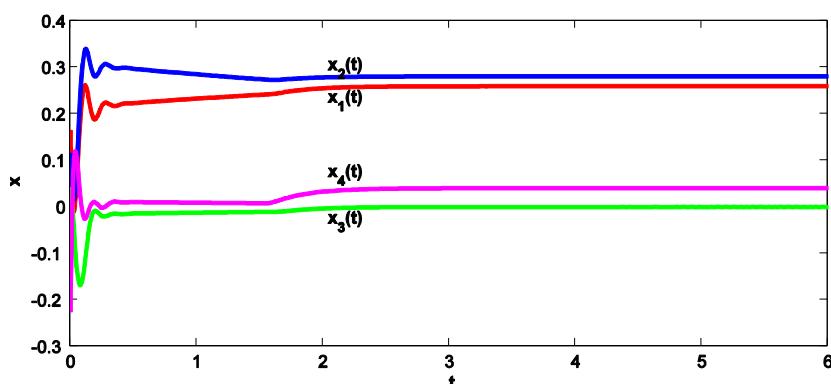
شکل ۳. رفتارهای گذراي x_1 ، x_2 ، x_3 و x_4 از شبکه عصبی (۱۶) در مثال (۳-۶).

مثال ۶-۴

$$\text{Min } f(x) = \dots / 4x_1 + x_1^r + x_1^r - x_1 x_r + \dots / \Delta x_r^r + \dots / \Delta x_r^r + \frac{x_1^r}{\gamma}.$$

$$\text{s.t.} \begin{cases} x_1 + x_r - x_r = \dots / \Delta, \\ x_1 + \dots / \Delta x_r - x_r = \dots / 4, \\ x \geq \dots \end{cases}$$

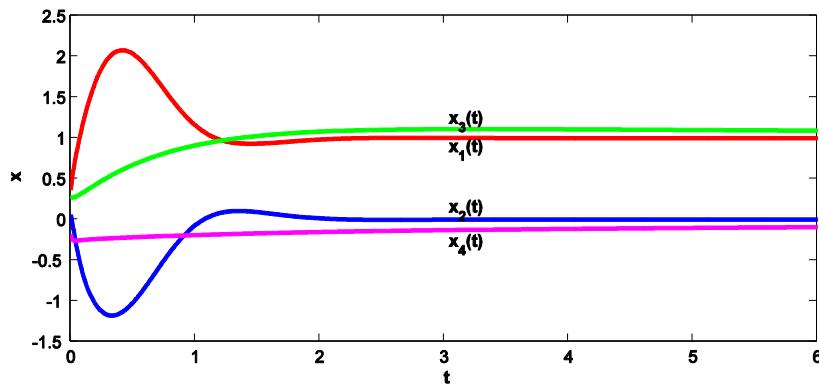
جواب بهینه $x^* = (0/40, 0/2595, 0/2809, 0/40)$ است. شکل ۴ نشان می‌دهد که مسیرها همگرا به جواب بهینه هستند. نتایج شبیه‌سازی نشان می‌دهند که نقطه شروع خارج ناحیه شدنی است؛ اما مسیرهای شبکه عصبی همگرا به جواب بهینه هستند.

شکل ۴. رفتارهای گذراي x_1 ، x_2 ، x_3 و x_4 از شبکه عصبی (۱۶) در مثال (۴-۶).

مثال ۵-۶

$$\begin{aligned} \text{Min } f(x) &= \frac{1}{2}(x_1^2 + x_2^2) + 2(x_3^2 + x_4^2) - \ln(x_1 x_2) + 3x_1 x_2 + 4x_3 x_4 - 2x_1 - 3x_2 \\ \text{s.t. } &\begin{cases} x_1 + x_2 = 1, \\ x_3 + x_4 = 1, \\ 0 \leq x_1 \leq 10, \quad 0 \leq x_2 \leq 10, \\ 0 \leq x_3 \leq 10, \quad 0 \leq x_4 \leq 10. \end{cases} \end{aligned}$$

شبیه‌سازی عددی نشان می‌دهد که مسیرهای خروجی از شبکه عصبی (۱۶) همگرا به جواب بهینه $x^* = (1, 0, 0, 0)^T$ هستند. به عنوان مثال شکل ۵ همگرایی را نشان می‌دهد.

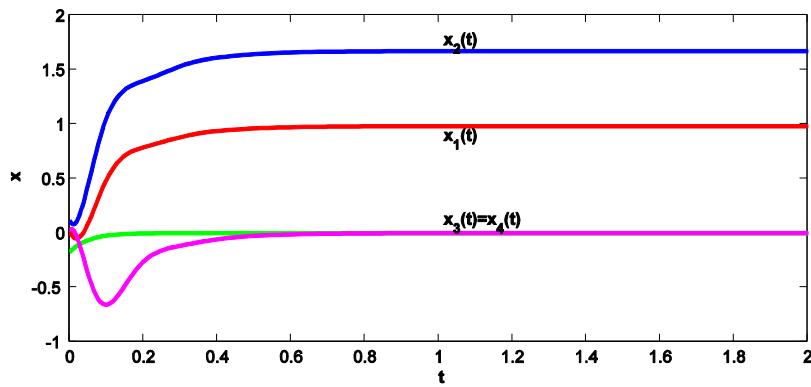


شکل ۵. رفتارهای گذراي x_1, x_2, x_3 و x_4 از شبکه عصبی (۱۶) در مثال (۵-۶).

مثال ۶-۷

$$\begin{aligned} \text{Min } f(x) &= 0.4x_1 + x_1^2 + x_2^2 - x_1 x_2 + 0.5x_3^2 + 0.5x_4^2 + \frac{1}{3}x_1^3 \\ \text{s.t. } &\begin{cases} -x_1 + x_2 - x_3 \leq 2, \\ 3x_1 + x_2 - x_3 - x_4 \leq 18, \\ \frac{1}{3}x_1 + x_2 - x_4 = 2, \\ x_1, x_2, x_3, x_4 \geq 0. \end{cases} \end{aligned}$$

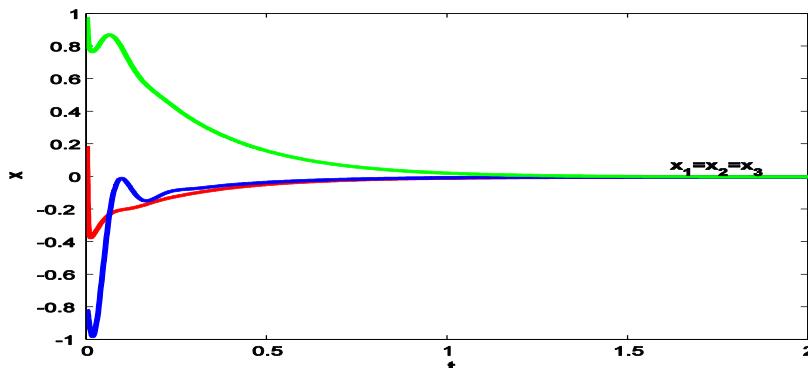
جواب بهینه برای این مساله $x^* = (0/9820, 1/6727, 0, 0)^T$ است. همگرایی مسیرهای گذراي $x(t)$ بر مبنای شبکه عصبی (۱۶) در شکل ۶ نشان داده می‌شود.

شکل ۶. رفتارهای گذراي x_1 ، x_2 ، x_3 و x_4 از شبکه عصبی (۱۶) در مثال (۶-۶).**مثال ۷**

$$\text{Min } f(x) = 11x_1^r + x_2^r + x_3^r + 2x_1x_2 + 8x_1x_3$$

$$\text{s.t.} \begin{cases} 2x_1 + 2x_2 + x_3 = 0, \\ -x_1 + x_2 \leq 1, \\ 3x_1 + x_3 \leq 4, \\ 0 \leq x_i \leq 6. \end{cases}$$

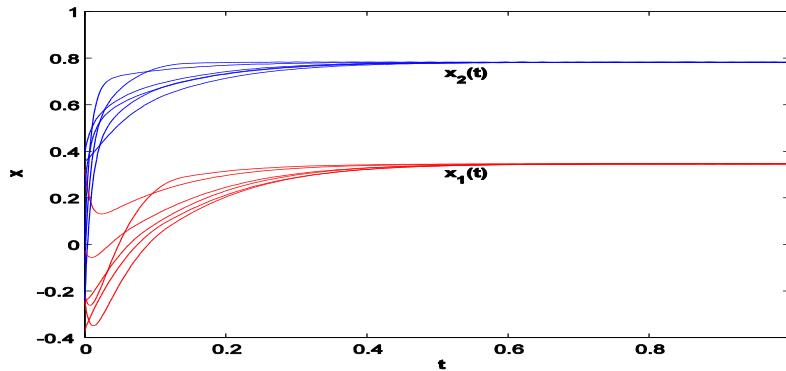
جواب بهینه $x^* = (0, 0, 0)^T$ است. همگرایی مسیرها به x^* در شکل ۷ نشان داده می‌شود.

شکل ۷. رفتارهای گذراي x_1 ، x_2 و x_3 از شبکه عصبی (۱۶) در مثال (۷-۶).**مثال ۸**

$$\text{Min } f(x) = \frac{1}{4}x_1^r + \frac{1}{2}x_1^r + \frac{1}{4}x_2^r + \frac{1}{2}x_3^r - 0.9x_1x_2$$

$$\text{s.t.} \begin{cases} x_1 + x_2 \leq 2, \\ -x_1 + x_2 \leq 2, \\ x_1 - 3x_2 \leq -2, \\ x_1, x_2 \geq 0. \end{cases}$$

قضیه ۴-۵ همگرایی سراسری شبکه عصبی پیشنهادی را به جواب بهینه $x^* = (0/3462, 0/7814)^T$ تضمین می‌کند. شکل ۸ همگرایی مسیرهای شبکه عصبی (۱۶) را با ۶ نقطه شروع متفاوت نشان می‌دهد.



شکل ۸ رفتارهای گذراي x_1 و x_2 از شبکه عصبی (۱۶) در مثال (۸-۶).

۷ نتیجه‌گیری

در این مقاله یک شبکه عصبی جدید برای حل مساله برنامه‌ریزی غیرخطی با محدودیت‌های هیبریدی ارایه شده است. شبکه عصبی پیشنهاد شده به هیچ پارامتر جریمه‌ای نیاز ندارد و ساختار آن نیز ساده است؛ بنابراین مدل پیشنهادی برای اجرا در سخت افزار بسیار مناسب است. به علاوه، نشان دادیم که مدل شبکه عصبی ارایه شده پایدار به مفهوم لیپانوف بوده و به جواب دقیق مساله اصلی همگرا می‌شود. از طرفی، به دلیل کاربرد وسیع مساله برنامه‌ریزی درجه دوم، مدل پیشنهادی و نتایج بدست آمده در این مقاله، هم از نظر تئوری و هم از نظر کاربرد بسیار مهم و قابل توجه می‌باشند. با ارایه چندین مثال نشان دادیم که مدل ارایه شده همگرای سراسری بوده و هیچ وابستگی به نقطه آغازین ندارد. نتایج شبیه‌سازی عددی نیز به روشنی همگرایی و خواص مدل پیشنهادی را نشان می‌دهند.

منابع

- [۸] ناصری، س.، طالشیان جلوه‌دار، ف.، تقی نژاد، ن.، خلیلی، ف.، (۱۳۹۱). مساله برنامه ریزی درجه دوم با ضرایب فازی: یک روش حل مبتنی بر اصل گسترش. مجله تحقیق در عملیات در کاربردهای آن، ۹(۳۵)، ۹-۲۵.
- [۵۳] اعتمادی، م.، (۱۳۸۶). حل مساله تخصیص تعیین یافته به روش شبکه عصبی. مجله تحقیق در عملیات در کاربردهای آن، ۴(۱۰)، ۹-۱۰.
- [1] Agrawal, S. K., Fabien, B. C. (1999). Optimization of Dynamic Systems, Kluwer Academic Publishers, Netherlands.
 - [2] Avriel, M. (1976). Nonlinear Programming: Analysis and Methods, Englewood Cliffs, NJ: Prentice-Hall.
 - [3] Bazaraa, M. S., Sherali, H. D., Shetty, C. M. (1993). Nonlinear Programming- Theory and Algorithms, 2nd ed. New York: Wiley.
 - [4] Bertsekas, D. P. (1989). Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, Englewood Cliffs, NJ.
 - [5] Cornuejols, G., Tutuncu, R. (2006). Optimization Methods in Finance, Cambridge University Press.
 - [6] Boyd, S., Vandenberghe, L. (2004). Convex Optimization, Cambridge University Press, Cambridge.

- [7] Fletcher, R. (1981). Practical Methods of Optimization, New York: Wiley.
- [9] Yoshikawa, T., (1990). Foundations of robotics: Analysis and control. Cambridge, MA: MIT Press.
- [10] Kalouptidis, N., (1997). Signal processing systems, theory and design. New York: Wiley.
- [11] Harker, P. T., Pang, J. S. (1990). Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms, and applications, *Math. Progr. ser. B* 48, 161-220.
- [12] He, B. S., Liao, L. Z. (2002). Improvements of some projection methods for monotone nonlinear variational inequalities, *Journal of Optimization Theory and Applications*, 112, 111-128.
- [13] Solodov, M. V., Tseng, P. (1996). A Modified projection-type methods for monotone variational inequalities, *SIAM Journal on Control Optimization*, 34, 1814-1830.
- [14] Tseng, P. (2000). A modified forward-backward splitting method for maximal monotone mappings, *SIAM Journal on Control Optimization*, 38, 431-446.
- [15] Cichocki, A., Unbehauer, R. (1993). Neural networks for optimization with bounded constraints, *IEEE Transactions on Neural Networks*, 4, 293-304.
- [16] McCulloch, W., Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.*, 5, 115-133.
- [17] Tank, D. W., Hopfield, J. J. (1986). Simple neural optimization networks: An A/D converter, signal decision circuit, and a linear programming pircuit, *IEEE Transactions on Circuits and Systems*, 33, 533-541.
- [18] Kennedy, M. P., Chua, L. O. (1988). Neural networks for nonlinear programming, *IEEE Transactions on Circuits and Systems*, 35, 554-562.
- [19] Cichocki, A., Unbehauer, R. (1993). Neural networks for optimization with bounded constraints, *IEEE Transactions on Neural Network*, 4, 293-304.
- [20] Effati, S., Nazemi, A. R. (2006). Neural network models and its application for solving linear and quadratic programming problems, *Applied Mathematics and Computation*, 172, 305-331.
- [21] Effati, S., Ghomashi, A., Nazemi, A.R. (2007). Application of projection neural network in solving convex programming problems, *Applied Mathematics and Computation*, 188, 1103-1114.
- [22] Forti, M., Nistri, P., Quincampoix, M. (2006). Convergence of neural networks for programming problems via a nonsmoothLojasiewicz inequality, *IEEE Transactions on Neural Networks*, 17, 1471-1486.
- [23] Friesz, T. L., Bernstein, D. H., Mehta, N. J., Tobin, R. L., Ganjlizadeh, S. (1994). Day-to-day dynamic network disequilibria and idealized traveler information systems. *Oper.ation Research*, 42, 1120–1136.
- [24] Gao, X. B., Liao, L.Z., Qi, L. Q. (2005). A novel neural network for variational inequalities with linear and nonlinear constraints, *IEEE Transactions on Neural Networks*, 16, 1305-1317.
- [25] Hu, X., (2009). Applications of the general projection neural network in solving extended linear-quadratic programming problems with linear constraints, *Neurocomputing*, 72 ,1131-1137.
- [26] Hu, X., Wang, J. (2007). Design of general projection neural networks for solving monotone linear variational inequalities and linear and quadratic optimization problems, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37, 1414-1421.
- [27] Liao, L.,Qi, H., Qi, L.(2001). Solving nonlinear complementarity problems with neural networks: a reformulation method approach, *Journal of Computational and Applied Mathematics*, 131, 343-359
- [28] Lillo, W. E., Loh, M. H., Hui, S., Z ā k, S. H. (1993). On solving constrained optimization problems with neural networks: A penalty method approach, *IEEE Transactions on Neural Networks*, 4, 931-939.
- [29] Liu, Q. S., Wang, J. (2008). A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming, *IEEE Transactions on Neural Networks*, 19, 558-570.
- [30] Maa, C. Y., Shanblatt, M. A. (1992). Linear and quadratic programming neural network analysis, *IEEE Transactions on Neural Networks*, 3, 580-594.
- [31] Malek, A., Hosseiniipour-Mahani, N., Ezazipour, S. (2010). Efficient recurrent neural network model for the solution of general nonlinear optimization problems, *Optimization Methods and Software*, 25, 1–18.
- [32] Nazemi, A. R. (2012). A dynamic system model for solving convex nonlinear optimization problems, *Communications in Nonlinear Science and Numerical Simulation*, 17, 1696-1705.
- [33] Nazemi, A. R., (2013). Solving general convex nonlinear optimization problems by an efficient neurodynamic model, *Engineering Applications of Artificial Intelligence*, 26, 685–696.
- [34] Arjmandzadeh, Z., Safi, M. R, Nazemi, A. R. (2017). A high performance neural network model for solving random interval linear programming problems, *Neural Networks*.

- [35] Nazemi, A. R., (2014). A neural network model for solving convex quadratic programming problems with some applications, *Engineering Applications of Artificial Intelligence*, 32, 54-62.
- [36] Nazemi, A. R., (2011). A dynamical model for solving degenerate quadratic minimax problems with constraints, *Journal of Computational and Applied Mathematics*, 236, 1282-1295.
- [37] Nazemi, A. R., Omidi, F. (2012). A capable neural network model for solving the maximum flow problem, *Journal of Computational and Applied Mathematics*, 236, 3498-3513.
- [38] Nazemi, A. R., Omidi, F.(2013). An efficient dynamic model for solving the shortest path problem, *Transportation Research Part C: Emerging Technologies*, 26, 1-19.
- [39] Nazemi, A. R., Effati, S. (2013). An application of a merit function for solving convex programming problems, *Computers and Industrial Engineering*, 66, 212—221.
- [40] Nazemi, A. R., Sharifi, E. (2013). Solving a class of geometric programming problems by an efficient dynamic model, *Communications in Nonlinear Science and Numerical Simulation*, 18, 692-709
- [41] Tao, Q., Cao, J., Xue, M., Qiao, H., Tao, Q., Cao, J., Sun, D. (2001). A high performance neural network for solving nonlinear programming problems with hybrid constraints, *Physics Letters A*, 288, 88–94.
- [42] Xia, Y., Wang, J. (2004). A general projection neural network for solving monotone variational inequality and related optimization problems, *IEEE Transactions on Neural Networks*, 15, 318-328.
- [43] Xia, Y., Wang, J. (2004). A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints, *IEEE Transactions on Circuits and Systems*, 51, 447-458
- [44] Xia, Y., Wang, J. (2000). On the stability of globally projected dynamical systems, *Journal of Optimization Theory and Applications*, 106, 129-150.
- [45] Xia, Y., Feng, G., Wang, J. (2004). A recurrent neural network with exponential convergence for solving convex quadratic program and related linear piecewise equation, *Neural Networks*, 17, 1003-1015
- [46] Xia, Y., Feng, G. (2005). An improved network for convex quadratic optimization with application to real-time beamforming, *Neurocomputing*, 64, 359-374
- [47] Xue, X., Bian, W. (2007). A project neural network for solving degenerate convex quadratic program, *Neurocomputing*, 70, 2449-2459.
- [48] Xue, X., Bian, W. (2009). A project neural network for solving degenerate quadratic minimax problem with linear constraints, *Neurocomputing*, 72, 1826-1838.
- [49] Yashtini, M., Malek, A. (2008). A discrete-time neural network for solving nonlinear convex problems with hybrid constraints, *Applied Mathematics and Computation*, 195, 576-584.
- [50] Yang, Y., Cao, J. (2006). Solving quadratic programming problems by delayed projection neural network, *IEEE Transactions on Neural Networks*, 17, 1630-1634.
- [51] Yang, Y., Cao, J. (2006). A delayed neural network method for solving convex optimization problems, *International Journal of Neural Systems*, 16, 295-303.
- [52] Yang, Y., Cao, J. (2008). A feedback neural network for solving convex constraint optimization problems, *Applied Mathematics and Computation*, 201, 340—350.
- [54] Ortega, T. M., Rheinboldt, W. C. (1970). *Iterative solution of nonlinear equations in Several variables*, Academic Press, New York.
- [55] Miller, R. K., Michel, A. N. (1982). *Ordinary Differential Equations*, Academic Press, New York.