

دسترسی در سایت <http://jnrm.srbiau.ac.ir>

سال پنجم، شماره نوزدهم، مرداد و شهریور ۱۳۹۸

شماره شاپا: ۲۵۸۸-۵۸۸۸

JNRM

پژوهش‌های نوین در ریاضی



دانشگاه آزاد اسلامی، واحد علوم و تحقیقات

## حل مسائل هدایت گرمایی تصادفی معکوس با استفاده از الگوریتم‌های بهینه‌سازی ازدحام ذرات و ژنتیک

ایرج حسین‌زاده شاه بلاقی<sup>۱</sup>، رضا پور قلی<sup>۲\*</sup>، حسن دانامزرعه<sup>۳</sup>، سید هاشم طبسی<sup>۴</sup>

(<sup>۱</sup>) دانشجو، دانشکده ریاضی و علوم کامپیوتر، دانشگاه دامغان

(<sup>۲</sup>) دانشیار، دانشکده ریاضی و علوم کامپیوتر، دانشگاه دامغان

(<sup>۳</sup>) مربی، دانشکده ریاضی و علوم کامپیوتر، دانشگاه دامغان

(<sup>۴</sup>) استادیار دانشکده ریاضی و علوم کامپیوتر، دانشگاه دامغان

تاریخ ارسال مقاله: ۹۷/۰۴/۰۴ تاریخ پذیرش مقاله: ۹۸/۰۳/۲۲

### چکیده

هدف اصلی در این مقاله، حل یک مسأله معادله دیفرانسیل جزئی تصادفی معکوس با استفاده از الگوریتم‌های تکاملی می‌باشد. الگوریتم ازدحام ذرات و الگوریتم ژنتیک، دو الگوریتم مورد نظر می‌باشند که در این مقاله از آن‌ها استفاده می‌شود. در این مقاله، برای حل مسأله معادله دیفرانسیل جزئی تصادفی معکوس، ابتدا با استفاده از روش کرانک-نیکلسون مسأله معکوس مورد نظر را گسسته سازی می‌کنیم، سپس با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات و الگوریتم ژنتیک به حل آن‌ها می‌پردازیم. الگوریتم‌هایی که در این مقاله ارائه شده است، نسبت به سایر روش‌های قدیمی که تا بحال ارائه شده است، مزیت‌هایی دارد. پیاده‌سازی این الگوریتم‌ها ساده‌تر است، زمان اجرای کمتری دارند و تقریب بهتری تولید می‌کنند. همچنین نتایج عددی بدست آمده در این مقاله نشان می‌دهند که جواب‌های بدست آمده برای مثال‌های مطرح شده در بخش نتایج عددی، دارای دقت بالایی هستند و خطای کمتری دارند. همه الگوریتم‌های موجود در این مقاله برای بدست آوردن نتایج عددی مورد نظر، بر روی پردازنده Pentium (R) Dual core E5700 با سرعت ۳،۰۰ گیگاهرتز پیاده‌سازی شده است.

**واژه‌های کلیدی:** معادله دیفرانسیل جزئی تصادفی، روش تفاضلات متناهی، مسائل معکوس، گسسته‌سازی، الگوریتم بهینه‌سازی ازدحام ذرات، الگوریتم ژنتیک.

$$x \in [0,1], \quad t \in [0,T], \quad (1)$$

که در آن  $\beta$ ، یک عدد تصادفی در بازه  $(0,1)$  می‌باشد. مسأله فوق، دارای شرط اولیه

$$u(x, 0) = \sin(\pi x), \quad x \in [0,1]$$

و شرایط مرزی

$$u(0,t) = u(1,t) = 0 \quad t \in [0,T],$$

می‌باشد. همچنین، جواب دقیق مسأله بصورت زیر است.

$$u(x, t) = e^{-\beta\pi^2 t} \sin(\pi x).$$

با اعمال روش تفاضلات متناهی بر روی معادله (۱)، داریم:

$$u_k^{n+1} = (1 - 2r)u_k^n + r(u_{k+1}^n + u_{k-1}^n)$$

$$u_k^0 = \sin(\pi x_k),$$

$$u_0^n = u_N^n = 0.$$

از آنجایی که

$$r = \frac{\beta\Delta t}{\Delta x^2}, t_n = n\Delta t, x_k = k\Delta x$$

داریم:

$$u_k^1 = \sin(k\pi\Delta x) \left(1 + 2r \sum_{i=1}^{\infty} (-1)^i \frac{(\pi\Delta x)^{2i}}{(2i)!}\right),$$

$$u_k^2 = \sin(k\pi\Delta x) \left(1 + 2r \sum_{i=1}^{\infty} (-1)^i \frac{(\pi\Delta x)^{2i}}{(2i)!}\right)^2,$$

$$u_k^3 = \sin(k\pi\Delta x) \left(1 + 2r \sum_{i=1}^{\infty} (-1)^i \frac{(\pi\Delta x)^{2i}}{(2i)!}\right)^3,$$

و سرانجام

$$u_k^n = \sin(k\pi\Delta x) \left(1 + 2r \sum_{i=1}^{\infty} (-1)^i \frac{(\pi\Delta x)^{2i}}{(2i)!}\right)^n.$$

در این مقاله، هدف حل مسائل معکوس می‌باشد نه مسائل مستقیم ولی در کل، حل مسائل مستقیم گامی برای حل مسائل معکوس می‌باشد.

## ۱- پیشگفتار

معادلات دیفرانسیل جزئی<sup>۱</sup> که به اختصار به آن‌ها PDE گفته می‌شود، نقش مهمی در توصیف پدیده‌ها در زمینه‌های مختلف علوم مانند دینامیک سیالات، مکانیک، اپتیک، انتقال حرارت، مغناطیس و ... ایفا می‌کنند [۷-۱]. این معادلات به‌طور کلی خطی یا غیرخطی هستند و برای حل آن‌ها راه‌حل‌های متفاوتی وجود دارند. گاهی اوقات، پارامتر و داده‌های اولیه این مسائل به علت ناقص بودن و کمبود اطلاعات، نامعلوم می‌باشند و در عمل، چنین سیستم‌هایی تحت اختلالات محیط اطراف قرار می‌گیرند و برای جبران کمبود اطلاعات و توصیف واقع گرایانه سیستم، یک Random Noise را در معادله معرفی می‌کنند که نتیجه آن، یک معادله دیفرانسیل جزئی تصادفی<sup>۲</sup> ایجاد می‌شود که به اختصار به آن‌ها RPDE گفته می‌شود. برخی از حوزه‌هایی که RPDEها به طور گسترده مورد استفاده قرار می‌گیرند عبارتند از: ریاضی، شیمی، فیزیک، مهندسی، زیست‌شناسی و ریاضی مالی. معمولاً برای حل RPDEها، از روش تفاضلات متناهی استفاده می‌کنیم [۶]. معادلات RPDE، در دو حالت قابل بررسی است. اگر در این مسائل هدف یافتن یک مجهول باشد آن‌گاه این مسائل جزء مسائل مستقیم حساب می‌شوند ولی اگر هدف از حل این مسائل، یافتن بیش از یک مجهول باشد آن‌گاه این مسائل جزء مسائل معکوس می‌باشند. در زمینه حل عددی مسائل معکوس، می‌توان به مراجع [۸-۱۱] اشاره نمود. نمونه‌هایی از مسائل مستقیم و معکوس را می‌توان به فرم زیر معرفی کرد.

### حالت اول (مسأله مستقیم): یک مسأله هدایت

گرمایی تصادفی مستقیم یک بعدی می‌تواند به فرم زیر معرفی گردد.

$$u_t(x, t) = \beta u_{xx}(x, t),$$

1. Partial Differential Equations
2. Random Partial Differential Equation

**ملاحظه ۱،۱.** برای حل مساله (۲) می‌توان گسسته سازی مساله را طوری در نظر گرفت که مقادیر  $u$  در تراز زمان  $t_{j+1}$  تنها با استفاده از مقادیر  $u$  در تراز زمان قبلی  $t_j$  محاسبه نشوند، بلکه به مقادیر  $u$  در تراز زمان  $t_{j+1}$  نیز بستگی داشته باشند. یک چنین روش‌هایی منتهی به روش‌های ضمنی خواهند شد، که ما در این مقاله از روش کرانک-نیکلسون<sup>۳</sup> استفاده می‌کنیم. روش‌های ضمنی در کل، بدون هیچ شرطی پایدار هستند.

**۱،۱. روش ضمنی کرانک-نیکلسون.** در این

روش، گسسته سازی معادله مساله (۲) به صورت زیر در نظر گرفته می‌شود.

$$\frac{u_{i,j+1} - u_{i,j}}{k} = \beta \left\{ \begin{array}{l} \frac{u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1}}{h^2} \\ - \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} \end{array} \right\}$$

که در آن از خطای حاصل صرف نظر شده است. با ساده سازی رابطه فوق داریم:

$$\begin{aligned} -ru_{i-1,j+1} + (2+2r)u_{i,j+1} - ru_{i+1,j+1} \\ = ru_{i-1,j} + (2-2r)u_{i,j} + ru_{i+1,j}, \\ u_{i,0} = f(ih), \\ u_{0,j} = 0, \\ u_{N,j} = 0, \end{aligned}$$

که در آن

$$r = \frac{\beta k}{h^2}, j \geq 0, i = 1, 2, \dots, N - 1$$

فرم ماتریسی معادله فوق نیز به صورت زیر می‌باشد:

**حالت دوم (مساله معکوس):** یک مساله هدایت گرمایی تصادفی معکوس یک بعدی می‌تواند به فرم زیر معرفی گردد.

$$u_t(x, t) = \beta u_{xx}(x, t), \quad 0 < x < 1, \quad 0 < t < t_m, \quad (2)$$

که در آن  $\beta$ ، یک عدد تصادفی در بازه (۰, ۱) می‌باشد. مساله فوق، دارای شرط اولیه

$$u_t(x, 0) = f(x), \quad 0 < x < 1$$

و شرایط مرزی زیر می‌باشد.

$$u_t(0, t) = p(t), \quad 0 < x < 1, \quad 0 < x < t_m$$

$$u_t(1, t) = q(t), \quad 0 < x < 1, \quad 0 < x < t_m$$

شرط اضافی مساله نیز بصورت زیر است.

$$U(a, t) = s(t), \quad 0 < t < t_m, \quad 0 < a < 1$$

در این مساله  $q(t)$  مجهول می‌باشد و هدف پیدا کردن جواب بهینه برای  $(u(x, t), q(t))$  است. برای حل مسائل معادلات دیفرانسیل جزئی تصادفی معکوس، روش‌های تحلیلی موجود نمی‌باشد و از این رو برای حل مسائل معادلات تصادفی معکوس از روش‌های عددی استفاده می‌شود. در این مقاله با استفاده از الگوریتم PSO<sup>۱</sup> و GA<sup>۲</sup>، مقدار بهینه‌ای برای  $q(t)$  بدست می‌آوریم.

بطور کلی برای حل این مسائل، با استفاده از تابع برازندگی (۳) که اختلاف مقدار واقعی (شرط اضافی) با مقدار بدست آمده از روش تفاضلات متناهی است، بصورت تصادفی انتخاب می‌شوند. در این معادله،  $u$  نشان دهنده مقدار بدست آمده از روش تفاضلات متناهی است و  $S$ ، همان شرط اضافی مساله است.

$$f(G) = \sum_{j=1}^m (u_j - s_j)^2 \quad (3)$$

سپس با استفاده از ترکیب اطلاعات محل فعلی‌اش و بهترین محلی که در گذشته در آن بوده است و همچنین اطلاعات یک یا چند ذره از بهترین ذرات موجود در جمع، جهتی را برای حرکت انتخاب می‌کند. همه ذرات، جهتی را برای حرکت انتخاب می‌کنند و پس از انجام حرکت، یک مرحله از الگوریتم به پایان می‌رسد. این مراحل چندین بار تکرار می‌شوند تا یک جواب بهینه به دست بیاید. در واقع الگوریتم انبوه ذرات که مقدار بهینه یک مساله را جستجو می‌کنند، همانند دسته‌ای از پرندگان عمل می‌کنند که به دنبال غذا می‌گردند [۲۱-۱۸].

اجزای الگوریتم بهینه‌سازی ازدحام ذرات عبارتند از:  
 الف)  $x = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$  مجموعه‌ای از مختصات است که موقعیت فعلی ذره را در فضای  $D$  بعدی نمایش می‌دهد. که ارزش موقعیت هر ذره  $i$ ،  $i = 1, 2, \dots, m$  با تابع برازندگی ارزیابی می‌شود و به صورت تصادفی ایجاد می‌شوند.

ب)  $x^{i,best}$  بهترین موقعیتی که ذره تا به حال تجربه کرده است (بهترین تجربه شخصی).

ج)  $x^{g,best}$  بهترین موقعیتی که به وسیله همه ذرات پیدا شده است (بهترین تجربه گروه).

د) ضریب اینرسی  $\omega$  روی همگرایی الگوریتم PSO تاثیر مستقیم دارد. مقدار زیاد برای  $\omega$  باعث می‌شود که ذرات موجود در الگوریتم، به جستجوی مناطق جدیدتر روی بیاورند و یک جستجوی سراسری را انجام دهند. در مقابل یک مقدار کم برای  $\omega$  باعث می‌شود که ذرات در منطقه‌ی محدودی بمانند و در واقع یک جستجوی محلی را انجام دهند. ما در اینجا مقدار اولیه ضریب اینرسی را برابر یک قرار دادیم و در هر تکرار به مقدار ۰.۹۹ ضرب کردیم. روش‌های گوناگونی برای مقدار دهی ضریب اینرسی وجود دارد.

ه)  $v_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$  سرعت حرکت ذره را نشان می‌دهد، که ما در اینجا سرعت اولیه مساله را صفر در نظر گرفتیم.

در هر تکرار سرعت و مکان ذره بروز می‌شود.

$$\begin{pmatrix} 2+2r & -r & 0 & 0 & 0 & 0 & 0 \\ -r & 2+2r & -r & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & -r & 2+2r & -r \\ 0 & 0 & 0 & 0 & 0 & -r & 2+2r \end{pmatrix} \times \begin{pmatrix} u_{1,j+1} \\ u_{2,j+1} \\ \vdots \\ u_{N-2,j+1} \\ u_{N-1,j+1} \end{pmatrix} = \begin{pmatrix} 2-2r & r & 0 & 0 & 0 & 0 & 0 \\ r & 2-2r & r & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & r & 2-2r & r \\ 0 & 0 & 0 & 0 & 0 & r & 2-2r \end{pmatrix} \times \begin{pmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{N-2,j} \\ u_{N-1,j} \end{pmatrix}$$

با حل دستگاه خطی به دست آمده،  $u_{ij}$ ها محاسبه می‌شوند.

## ۲- الگوریتم بهینه‌سازی ازدحام ذرات برای حل مسائل هدایت گرمایی تصادفی معکوس

جیمز کندی<sup>۱</sup>، روانشناس اجتماعی، و راسل سی ابرهارت<sup>۲</sup>، مهندس برق، صاحبان اصلی ایده‌ی الگوریتم PSO می‌باشند. آن‌ها در ابتدا قصد داشتند که با بهره‌گیری از مدل‌های اجتماعی و روابط موجود اجتماعی، نوعی از هوش محاسباتی را به وجود بیاورند که به توانایی‌های فردی ویژه نیازی نداشته باشد. اولین شبیه سازی آن‌ها در سال ۵۱۹۹ انجام گردید [۱۲و۷].

کار کندی و ابرهارت، منجر به ایجاد الگوریتمی قوی برای بهینه‌سازی، به نام الگوریتم بهینه‌سازی انبوه ذرات یا PSO شدند [۱۷-۱۳]. در الگوریتم PSO، تعدادی از موجودات وجود دارند، که به آن‌ها ذره گفته می‌شود و در فضای جستجوی تابعی که قصد بهینه کردن مقدار آن را داریم، پخش شده‌اند. هر ذره مقدار تابع هدف را در موقعیتی از فضا که در آن قرار گرفته است، محاسبه کند.

1. James Kennedy
2. Russel C. Eberhart

اثبات مربوط به همگرایی الگوریتم انبوه ذرات در مراجع [۲۳ و ۲۲] آورده شده است.

### ۳- الگوریتم ژنتیک برای حل مسایل هدایت معکوس گرمایی تصادفی

ایده اصلی الگوریتم ژنتیک توسط جان هلند<sup>۱</sup> مطرح شد [۱۷]. الگوریتم‌های ژنتیک از اصول انتخاب طبیعی داروین برای یافتن فرمول بهینه جهت پیش‌بینی یا تطبیق الگو استفاده می‌کنند. ایده اساسی این الگوریتم انتقال خصوصیات موروثی توسط ژن‌ها است [۲۴ و ۲۵]. فرض کنید مجموعه خصوصیات انسان توسط کروموزوم‌های او به نسل بعدی منتقل می‌شوند. هر ژن در این کروموزوم‌ها نماینده یک خصوصیت است. بعنوان مثال ژن ۱ می‌تواند رنگ مو باشد، ژن ۲ طول قد، ژن ۳ رنگ چشم و الی آخر. حال اگر این کروموزوم به تمامی به نسل بعد انتقال یابد، تمامی خصوصیات نسل بعدی شبیه به خصوصیات نسل قبل خواهد بود. بدیهی است که در عمل چنین اتفاقی رخ نمی‌دهد. در واقع بصورت همزمان دو اتفاق برای کروموزوم‌ها می‌افتد. اتفاق اول جهش است. «جهش»<sup>۲</sup> به این صورت است که بعضی ژن‌ها بصورت کاملاً تصادفی تغییر می‌کنند. البته تعداد این گونه ژن‌ها بسیار کم می‌باشد اما در هر حال این تغییر تصادفی همانگونه که پیشتر دیدیم بسیار مهم است. مثلاً ژن رنگ چشم می‌تواند بصورت تصادفی باعث شود تا در نسل بعدی یک نفر دارای چشمان سبز باشد. در حالی که تمامی نسل قبل دارای چشم قهوه‌ای بوده‌اند. علاوه بر جهش اتفاق دیگری که می‌افتد و البته این اتفاق به تعداد بسیار بیشتری نسبت به جهش رخ می‌دهد چسبیدن دو کروموزوم از طول به یکدیگر و تبادل برخی قطعات بین دو کروموزوم است. این مساله با نام ترکیب<sup>۳</sup> شناخته می‌شود. این همان چیزی است که باعث می‌شود تا فرزندان ترکیب ژن‌های متفاوتی را (نسبت به والدین خود) به فرزندان خود انتقال دهند.

$$v_{i,d}[k+1] = \omega v_{i,d}[k] + c_1 r_1 (x_d^{i,best} - x_{i,d}[k]) + c_2 r_2 (x_d^{g,best}[k] - x_{i,d}[k]), \quad (۴)$$

$$x_{i,d}[k+1] = x_{i,d}[k] + v_{i,d}[k+1], \quad (۵)$$

که در آن  $d = 1, 2, \dots, D$ ،  $i = 1, 2, \dots, m$  و  $r_1$  و  $r_2$  اعدادی تصادفی در بازه  $[0, 1]$  می‌باشند. بعلاوه  $c_1$  و  $c_2$  ضرایب یادگیری هستند.  $c_1$  ضریب یادگیری مربوط به تجارب شخص هر ذره است و درمقابل  $c_2$  ضریب یادگیری مربوط به تجارب کل جمع می‌باشد. مراحل اجرای الگوریتم PSO با مقادیر حقیقی را می‌توان به صورت زیر در نظر گرفت.

**مرحله ۱:** تنظیم پارامترهای موجود در الگوریتم از قبیل ساینز جمعیت (تعداد ذرات)، ماکزیمم سرعت ( $V_{max}$ )، ضریب اینرسی ( $w$ )، دو ضریب یادگیری ( $c_1$ ،  $c_2$ ) و دو متغیر تصادفی ( $r_1$ ،  $r_2$ ) که در بازه  $[0, 1]$  هستند.

**مرحله ۲:** مقداردهی اولیه موقعیت و سرعت هر ذره به صورت تصادفی.

**مرحله ۳:** مقدار دهی اولیه به صورت یک عدد تصادفی در بازه  $(0, 1)$ .

**مرحله ۴:** محاسبه تابع برازندگی برای هر ذره.

**مرحله ۵:** به روزرسانی  $x^{i,best}$  برای هر ذره. برای هر ذره، اگر مقدار برازندگی کمتر شده است این مقدار بعنوان مقدار جدید بهترین موقعیتی که ذره تا به حال تجربه کرده است در نظر گرفته می‌شود.

**مرحله ۶:** به روزرسانی  $x^{g,best}$  و  $x^{i,best}$  ذره‌ای که کمترین مقدار برازندگی را بین همه ذرات دارد.

**مرحله ۷:** به روزرسانی سرعت هر ذره مطابق فرمول (۴).

**مرحله ۸:** به روزرسانی موقعیت هر ذره مطابق فرمول (۵).

**مرحله ۹:** مرحله ۴ تا ۸ را تا زمانی که جواب قابل قبولی پیدا کنیم یا به ماکزیمم تکرارها برسیم تکرار می‌کنیم.

1. John Holland
2. Mutation
3. Crossover

**مرحله ی ۱:** تنظیم پارامترهای موجود در الگوریتم از قبیل سائز جمعیت (تعداد کروموزم‌ها)، ماکزیمم تعداد جهش ژن و عمل ترکیب.

**مرحله ی ۲:** مقداردهی اولیه کروموزوم‌ها به صورت تصادفی.

**مرحله ی ۳:** مقدار دهی اولیه  $\beta$  به صورت یک عدد تصادفی در بازه  $(0,1)$ .

**مرحله ی ۴:** محاسبه تابع برازندگی برای هر کروموزوم.

**مرحله ی ۵:** عمل انتخاب.

**مرحله ی ۶:** عمل ترکیب.

**مرحله ی ۷:** عمل جهش.

**مرحله ی ۸:** مرحله ۴ تا ۷ را تا زمانی که جواب قابل قبولی پیدا کنیم یا به ماکزیمم تکرارها برسیم تکرار می‌کنیم.

اثبات مربوط به همگرایی الگوریتم ژنتیک در مقاله [۲۶] آورده شده است.

#### ۴- نتایج عددی

هدف اصلی این بخش نشان دادن توانایی الگوریتم‌های GA و PSO برای حل مسائل مربوط به هدایت گرمایی تصادفی معکوس است. برای انجام این کار، سه مثال از مسأله هدایت گرمایی تصادفی معکوس با استفاده از الگوریتم‌های GA و PSO حل می‌شوند و سپس نتایج بدست آمده مورد بررسی قرار می‌گیرند.

**ملاحظه ۴.۱.** در یک مسأله هدایت گرمایی معکوس دو منبع خطا در تقریب شرط و یا شرایط مجهول مسأله وجود دارد. منبع اول انحراف اجتناب ناپذیر از جواب می‌باشد و منبع دوم خطا، در اندازه گیری شرط و یا شرایط اضافی می‌باشد. این دو منبع خطا، باعث به وجود آمدن خطای کلی می‌شود که به صورت زیر محاسبه می‌شود:

$$s = \left[ \frac{1}{N-1} \sum_{i=1}^N (\hat{f}_i - f_i)^2 \right]^{\frac{1}{2}}, \quad (6)$$

الگوریتم ژنتیک می‌تواند برای حل مسایل بهینه‌سازی مورد استفاد قرار گیرد، حتی می‌توان از این روش برای حل مسائل معادلات دیفرانسیل استفاده کرد.

اجزای تشکیل دهنده الگوریتم ژنتیک عبارتند از:

#### (الف) کروموزوم (جمعیت اولیه)

$G_i = \{g_{i,1}, g_{i,2}, \dots, g_{i,n}\}$  برای  $i = 1, 2, \dots, n$  به صورت تصادفی انتخاب می‌شوند، که ارزش هر کروموزوم توسط تابع برازندگی ارزیابی می‌شود.

#### (ب) انتخاب (Selection)

در این مرحله، دو والد جهت جفت‌گیری و تولید کروموزوم جدید انتخاب می‌شود. که روش‌های گوناگونی موجود است، ولی ما در اینجا از روش چرخ رولت استفاده می‌کنیم.

#### (ج) عمل ترکیب (Crossover)

در این روش ژن‌های فرزندان به صورت تصادفی و با یک احتمال از والدین انتخاب می‌شود. ما از روش انتخاب یک نقطه به صورت تصادفی در طول دو کروموزوم که به عنوان والدین انتخاب شده‌اند و جابجایی کروموزوم‌ها از آن نقطه استفاده کردیم.

**مثال:** اگر  $g_1 = 00000$  و  $g_2 = 11111$  در عمل ترکیب اگر نقطه تصادفی ۲ باشد، فرزندان جدید به صورت  $g'_1 = 11000$  و  $g'_2 = 00111$  می‌باشند.

#### (د) جهش (Mutation)

به پیدایش هر نوع تغییر در ژن‌های یک کروموزوم جهش می‌گوییم که به صورت تصادفی تعداد محدودی از کروموزوم‌ها انتخاب و عمل جهش در ژن به صورت تصادفی انجام می‌شود. در این جا ما از روش جهش حقیقی<sup>۱</sup> استفاده کردیم.

**(ه) ارزش گذاری** کروموزوم‌ها با تابع برازندگی و تکرار عمل **ب** تا **د** تا زمان برقراری شرط مسأله که شرط مسأله در اینجا تعداد تکرار می‌باشد.

مراحل اجرای الگوریتم GA با مقادیر حقیقی را می‌توان به صورت زیر در نظر گرفت.

#### 1. Real Mutation

$$u(1,t) = q(t), \quad 0 < t \leq 1,$$

که در آن  $\beta$  یک عدد تصادفی در بازه (۰ و ۱) می‌باشد. همچنین شرط اضافی مساله به صورت زیر است.

$$u(0.05, t_i) = s(t_i),$$

$$t_i = i \times 0.05, \quad i = 1, 2, 3, \dots, 20$$

که مقدار واقعی  $q(t)$  نیز برابر است با

$$q(t) = e^{-\beta t} \cos(1)$$

در این جا  $N$  تعداد نقاط کلی برای محاسبه‌ی خطای کلی،  $\hat{f}_i$  مقدار به دست آمده از تابع درون یابی شده و  $f_i$  مقدار واقعی پارامتر مورد نظر می‌باشد. [23]

**مثال ۱.۴.** مساله هدایت گرمایی معکوس زیر را در نظر می‌گیریم.

$$u_t(x, t) = \beta u_{xx}(x, t),$$

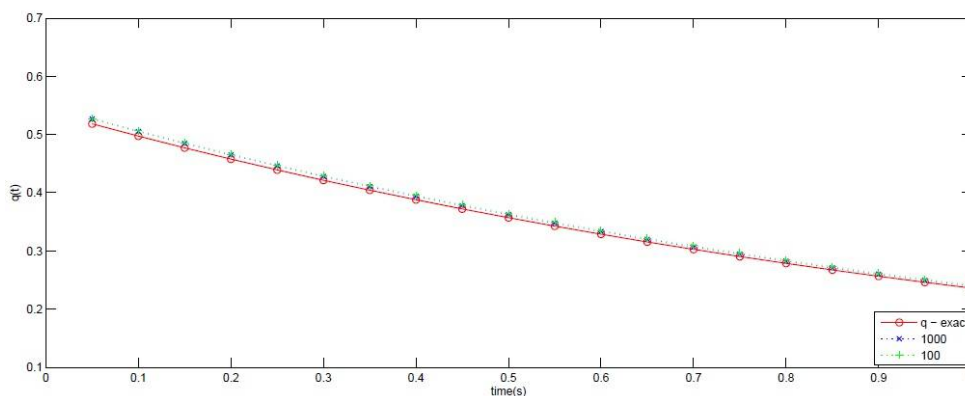
$$0 < x < 1, \quad 0 < t < t_m,$$

$$u(x, 0) = \cos(x), \quad 0 < t \leq 1,$$

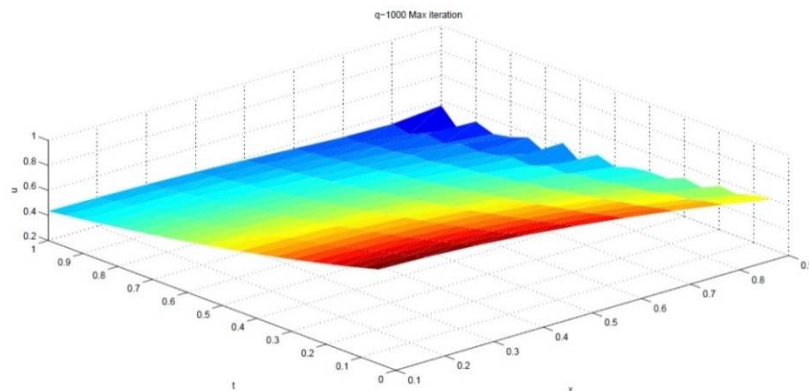
$$u(0, t) = e^{-\beta t}, \quad 0 < t \leq 1,$$

### نتایج حاصل از الگوریتم PSO:

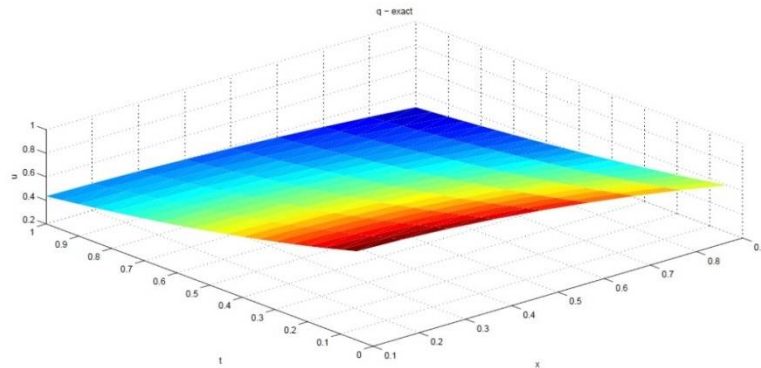
شکل ۱. نمودار  $q(t)$  واقعی و  $q(t)$  تقریب زده شده توسط الگوریتم PSO برای ۱۰۰ و ۱۰۰۰ تکرار با مقادیر حقیقی، برای ۱۰۰ ذره و  $\beta = ۰.۸۲۷۷۹$  برای مثال ۱.۴.



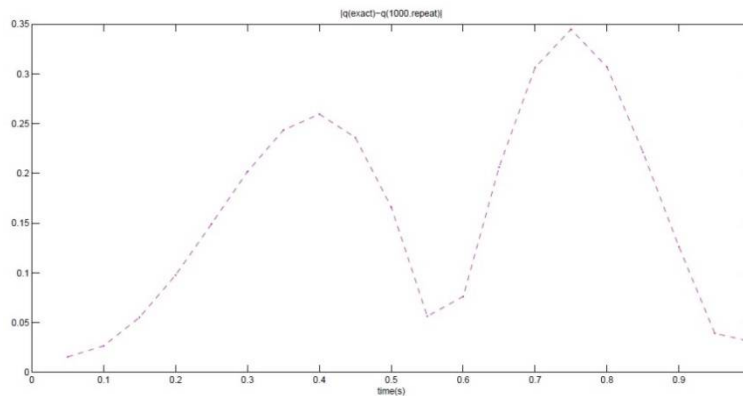
شکل ۲. شکل بدست آمده برای  $U(x, t)$  توسط  $q(t)$  بدست آمده از الگوریتم PSO برای ۱۰۰۰ تکرار و  $\beta = ۰.۸۲۷۷۹$  برای مثال ۱.۴.



شکل ۳. شکل بدست آمده برای  $U(x, t)$  توسط مقادیر واقعی  $q(t)$  و  $\beta = 0.82779$  برای مثال ۱.۴.



شکل ۴. قدر مطلق تفاضل  $q(t)$  واقعی و  $q(t)$  بدست آمده از الگوریتم PSO برای ۱۰۰۰ تکرار برای مثال ۱.۴.



جدول ۱. نتایج اجرای الگوریتم PSO با مقادیر حقیقی برای پیدا کردن  $q(t)$  مجهول با ۱۰۰ و ۱۰۰۰ تکرار برای مثال ۱.۴.

$\beta$	Time (s)	S	بهترین برازندگی	ماکزیمم تکرار
۰/۸۲۷۷۹	۱۰/۱۹۸۶	۰/۰۲۵۵	$5/1042 \times 10^{-5}$	۱۰۰
۰/۸۲۷۷۹	۹۲/۳۷۵۴۰۲	۰/۰۰۲	$2/6807 \times 10^{-6}$	۱۰۰۰

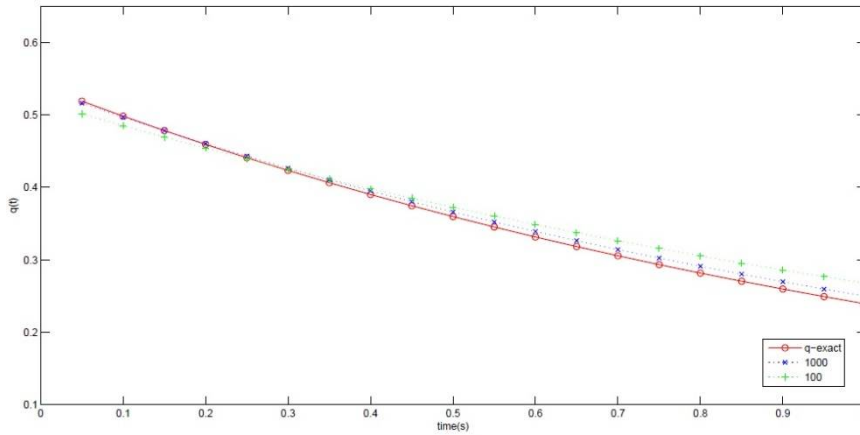
جدول ۲. مقادیر  $q(t)$  بدست آمده توسط الگوریتم PSO برای ۱۰۰ و ۱۰۰۰ تکرار برای مثال ۱.۴.

ذره							تکرار
$p_7$	$p_6$	$p_5$	$p_4$	$p_3$	$p_2$	$p_1$	۱۰۰
۰/۳۹۴	۰/۴۹۹	۰/۳۶۱	۰/۴۸۹	۰/۴۹۷	۰/۴۱۶	۰/۶۳۴	
$p_{14}$	$p_{13}$	$p_{12}$	$p_{11}$	$p_{10}$	$p_9$	$p_8$	
۰	۰/۵۲۸	۰/۲۲۸	۰/۳۷۵	۰/۳۳۳	۰/۴۲۶	۰/۳۴۳	
$p_{20}$	$p_{19}$	$p_{18}$	$p_{17}$	$p_{16}$	$p_{15}$		
۰/۳۰۲	۰/۳۳۷	۰/۰۹۳	۰/۴۶۰	۰/۰۵۳	۰/۵۷۵		
$p_7$	$p_6$	$p_5$	$p_4$	$p_3$	$p_2$	$p_1$	۱۰۰۰
۰/۳۸۵	۰/۴۳۲	۰/۴۴۷	۰/۴۲۳	۰/۵۴۲	۰/۴۰۸	۰/۶۲۰	
$p_{14}$	$p_{13}$	$p_{12}$	$p_{11}$	$p_{10}$	$p_9$	$p_8$	
۰	۰/۵۷۰	۰/۱۴۷	۰/۴۵۰	۰/۳۱۲	۰/۳۷۷	۰/۴۰۴	
$p_{20}$	$p_{19}$	$p_{18}$	$p_{17}$	$p_{16}$	$p_{15}$		
۰/۲۹۷	۰/۳۰۸	۰/۱	۰/۵	۰/۰۶	۰/۶		

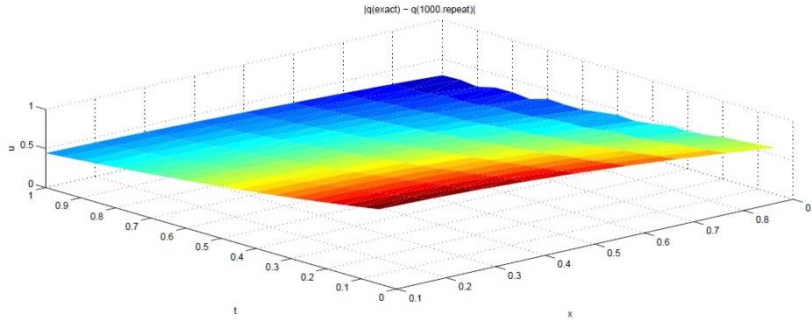


نتایج حاصل از الگوریتم GA:

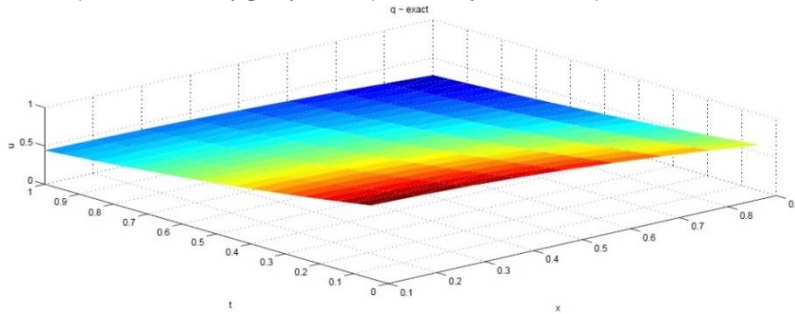
شکل ۵. نمودار  $q(t)$  واقعی و  $q(t)$  تقریب زده شده توسط الگوریتم GA برای ۱۰۰ و ۱۰۰۰ تکرار با مقادیر حقیقی، برای ۱۰۰ کروموزوم و  $\beta = ۰.۸۱۴۷$  برای مثال ۱.۴.



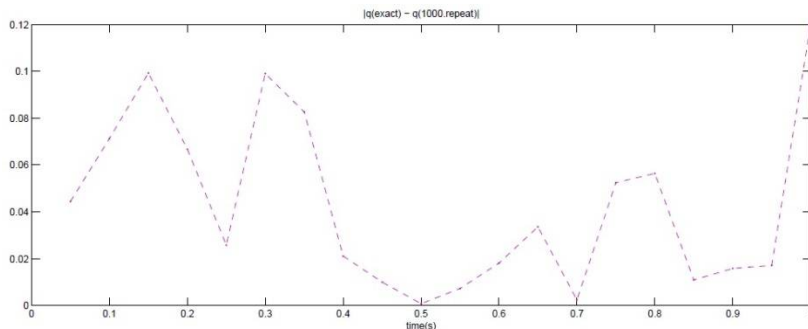
شکل ۶. شکل بدست آمده برای  $U(x, t)$  توسط  $q(t)$  بدست آمده از الگوریتم GA برای ۱۰۰۰ تکرار و  $\beta = ۰.۸۱۴۷$  برای مثال ۱.۴.



شکل ۷. شکل بدست آمده برای  $U(x, t)$  توسط مقادیر با  $q(t)$  واقعی و  $\beta = ۰.۸۱۴۷$  برای مثال ۱.۴.



شکل ۸. قدر مطلق تفاضل  $q(t)$  واقعی و  $q(t)$  بدست آمده از الگوریتم GA با ۱۰۰۰ تکرار برای مثال ۱.۴.



جدول ۳. نتایج اجرای الگوریتم GA با پیدا کردن  $q(t)$  مجهول برای ۱۰۰ و ۱۰۰۰ تکرار برای مثال ۱.۴.

تکرار	بهترین برازندگی	S	Time (s)	$\beta$
۱۰۰	$9/9977 \times 10^{-5}$	۰/۰۰۵	۱۰/۰۶۷۸	۰/۸۱۴۷
۱۰۰۰	$2/0422 \times 10^{-5}$	۰/۰۰۲	۹۲/۹۵۹۶	۰/۸۱۴۷

جدول ۴. مقادیر  $q(t)$  بدست آمده توسط الگوریتم GA برای ۱۰۰ و ۱۰۰۰ تکرار برای مثال ۱.۴.

تکرار	کروموزوم					
	p۱	p۲	p۳	p۴	p۵	p۶
۱۰۰	۰/۵۷۹	۰/۴۴۶	۰/۵۳۳	۰/۴۲۵	۰/۴۴۹	۰/۴۹۳
	۰/۳۱۶	۰/۴۱۱	۰/۳۸۳	۰/۳۳۳	۰/۲۶۳	۰/۳۹۵
	۰/۲۲۸	۰/۳۶۴	۰/۲۷۸	۰/۱۸۳	۰/۲۵۱	۰/۵۱۷
۱۰۰۰	۰/۵۶۷	۰/۴۴۶	۰/۵۲۹	۰/۴۲۴	۰/۴۶۲	۰/۳۷۵
	۰/۳۱۶	۰/۴۱۳	۰/۳۴۲	۰/۳۸۲	۰/۲۶۳	۰/۳۷۷
	۰/۲۵۵	۰/۳۰۹	۰/۲۸۸	۰/۱۹۶	۰/۳۱۵	۰/۲۱۸

در مثال ۱.۴. برای پیدا کردن  $q(t)$  مجهول، الگوریتم‌های ژنتیک و ازدحام ذرات با تکرارهای برابر، اجرا شدند. نتایج بدست آمده از اجرای این الگوریتم‌ها در جداول ۱ و ۳ نشان داده شده اند. مقایسه این نتایج بیانگر این مطلب است که الگوریتم ازدحام ذرات با زمان تقریباً برابر نتایج بهتری نسبت به الگوریتم ژنتیک ایجاد کرده است. زیرا مقدار خطای کلی (S) در الگوریتم ازدحام ذرات نسبت به ژنتیک کمتر است. همچنین شکل های ۲ و ۶ بدرتیب مربوط به  $U(x, t)$  بدست آمده با  $q(t)$  تقریب زده شده بوسیله الگوریتم ازدحام ذرات و الگوریتم ژنتیک می‌باشند. نوسان ایجاد شده در این اشکال در محدوده  $X=0,9$  به دلیل خطای اندکی است که در تقریب  $q(t)$  مجهول وجود دارد. شکل های ۱ و ۵ بدرتیب نمودارهای  $q(t)$  واقعی و مجهول تقریب زده شده بوسیله الگوریتم ازدحام ذرات و ژنتیک را نشان می‌دهند. در این اشکال وقتی تعداد تکرار از ۱۰۰ به ۱۰۰۰ تغییر

در مثال ۱.۴. برای پیدا کردن  $q(t)$  مجهول، الگوریتم‌های ژنتیک و ازدحام ذرات با تکرارهای برابر، اجرا شدند. نتایج بدست آمده از اجرای این الگوریتم‌ها در جداول ۱ و ۳ نشان داده شده اند. مقایسه این نتایج بیانگر این مطلب است که الگوریتم ازدحام ذرات با زمان تقریباً برابر نتایج بهتری نسبت به الگوریتم ژنتیک ایجاد کرده است. زیرا مقدار خطای کلی (S) در الگوریتم ازدحام ذرات نسبت به ژنتیک کمتر است. همچنین شکل های ۲ و ۶ بدرتیب مربوط به  $U(x, t)$  بدست آمده با  $q(t)$  تقریب زده شده بوسیله الگوریتم ازدحام ذرات و الگوریتم ژنتیک می‌باشند. نوسان ایجاد شده در این اشکال در محدوده  $X=0,9$  به دلیل خطای اندکی است که در تقریب  $q(t)$  مجهول وجود دارد. شکل های ۱ و ۵ بدرتیب نمودارهای  $q(t)$  واقعی و مجهول تقریب زده شده بوسیله الگوریتم ازدحام ذرات و ژنتیک را نشان می‌دهند. در این اشکال وقتی تعداد تکرار از ۱۰۰ به ۱۰۰۰ تغییر

یافته است نمودار  $q(t)$  تقریب زده شده به نمودار  $q(t)$  می‌گیریم. واقعی نزدیکتر شده است زیرا خطای کلی کمتر شده است. همچنین اشکال ۴ و ۸ نمودارهای خطای ایجاد شده در تکرار ۱۰۰۰ برای الگوریتم‌های ازدحام ذرات و ژنتیک را نشان می‌دهند. این نمودارها نشان می‌دهند که وقتی تکرار زیاد شده و مقادیر مربوط به  $q(t)$  مجهول به  $q(t)$  واقعی نزدیک می‌شود نقاط متوالی تقرب زده شده برای  $q(t)$  حول نقاط واقعی  $q(t)$  نوسان می‌کنند. در نهایت جداول ۲ و ۴ بترتیب مقادیر تقریب زده شده در نقاط  $x = 0.1, 0.15, 0.20, \dots, 1$  برای  $q(t)$  را در تکرارهای ۱۰۰ و ۱۰۰۰ را برای الگوریتم‌های ازدحام ذرات و ژنتیک را نشان می‌دهند.

$$u_t(x, t) = \beta u_{xx}(x, t),$$

$$0 < x < 1, \quad 0 < t < 1$$

$$u(x, 0) = \sin x^2 + 1, \quad 0 < t \leq 1$$

$$u(0, t) = 2\beta t + 1, \quad 0 < t \leq 1$$

$$u(1, t) = q(t), \quad 0 < t \leq 1,$$

که در آن  $\beta$  یک عدد تصادفی در بازه (۰،۱) می‌باشد. همچنین شرط اضافی مساله به صورت زیر است.

$$u(0.05, t_i) = s(t_i),$$

$$i = 1, 2, 3, \dots, 20 \quad t_i = i \times 0.05,$$

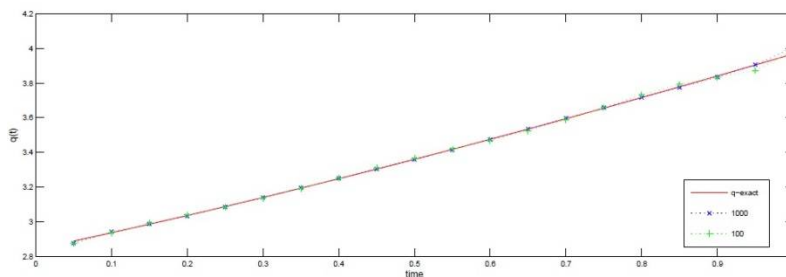
که مقدار واقعی  $q(t)$  نیز برابر است با

$$q(t) = e^{-\beta t} \sin(1) + 2\beta t + 2$$

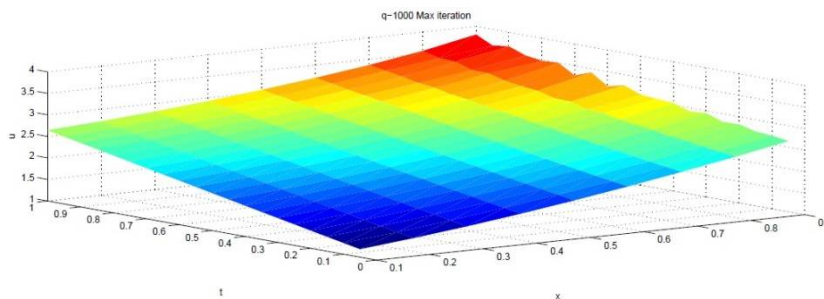
مثال ۲.۴. مساله هدایت گرمایی معکوس زیر را در نظر

### نتایج حاصل از الگوریتم PSO:

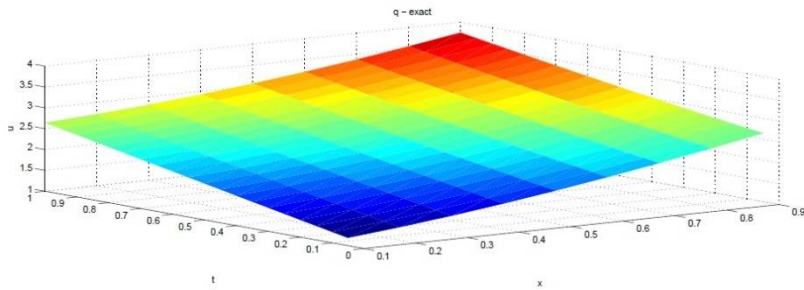
شکل ۹. نمودار  $q(t)$  واقعی و  $q(t)$  تقریب زده شده توسط الگوریتم PSO برای ۱۰۰ و ۱۰۰۰ تکرار با مقادیر حقیقی، برای ۱۰۰ ذره و  $\beta = ۰.۷۹۴۴$  برای مثال ۲.۴.



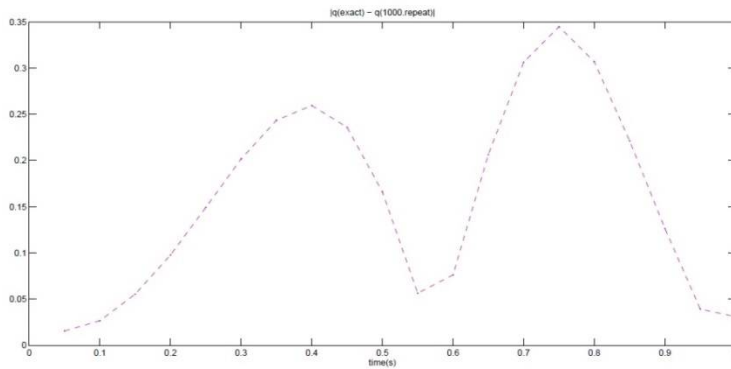
شکل ۱۰. شکل ایجاد شده  $q(t)$  توسط مقادیر بدست آمده از الگوریتم PSO برای ۱۰۰۰ تکرار و  $\beta = ۰.۷۹۴۴$  برای مثال ۲.۴.



شکل ۱۱. شکل ایجاد شده توسط مقادیر واقعی برای  $q(t)$  و  $\beta = 0.7944$  برای مثال ۲.۴.



شکل ۱۲. قدر مطلق تفاضل  $q(t)$  واقعی و  $q(t)$  بدست آمده از الگوریتم PSO برای ۱۰۰۰ تکرار برای مثال ۲.۴.



جدول ۵. نتایج اجرای الگوریتم PSO با مقادیر حقیقی برای پیدا کردن شرط مجهول  $q(t)$  برای ۱۰۰ و ۱۰۰۰ تکرار برای مثال ۲.۴.

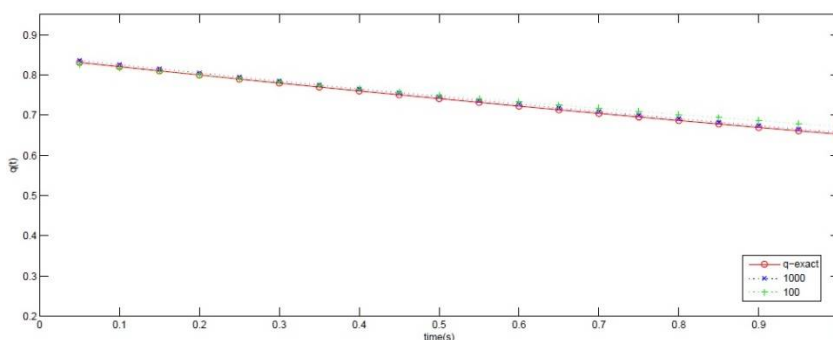
$\beta$	Time (s)	S	بهترین برازندگی	تکرار
۰/۷۹۴۴	۹/۵۰۵۲	۰/۰۲۷۵	$۸/۲۹۲۲ \times 10^{-۵}$	۱۰۰
۰/۷۹۴۴	۳۹/۵۶۲۱	۰/۰۲۸۰	$۹/۲۶۶۲ \times 10^{-۶}$	۱۰۰۰

جدول ۶. مقادیر  $q(t)$  بدست آمده توسط الگوریتم PSO برای ۱۰۰ و ۱۰۰۰ تکرار برای مثال ۲.۴.

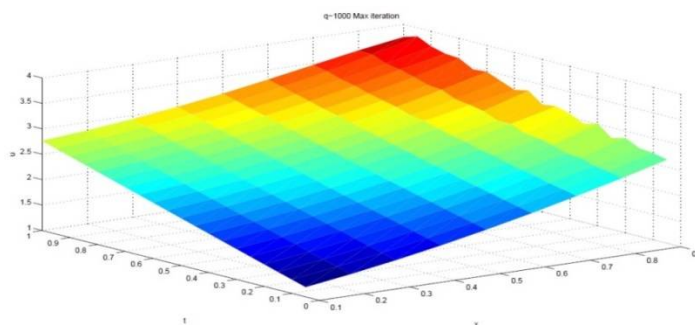
تکرار						
ذره						
$p_7$	$p_6$	$p_5$	$p_4$	$p_3$	$p_2$	$p_1$
۲/۹۵	۳/۳۴۵	۲/۹۵۵	۳/۱۰۲	۲/۹۵	۲/۹۶	۲/۸۷
$p_{14}$	$p_{13}$	$p_{12}$	$p_{11}$	$p_{10}$	$p_9$	$p_8$
۳/۳۷	۳/۶	۳/۵۱۵	۳/۳۱۶	۳/۵۲۳	۳/۰۸۵	۳/۵
$p_{20}$	$p_{19}$	$p_{18}$	$p_{17}$	$p_{16}$	$p_{15}$	
۴	۳/۸۱	۳/۹۱۷	۳/۸۵۵	۳/۴۵	۴	
$p_7$	$p_6$	$p_5$	$p_4$	$p_3$	$p_2$	$p_1$
۲/۹۵	۳/۳۴۱	۲/۹۴	۳/۱۳۳	۲/۹۳	۲/۹۶۲	۲/۸۷۳
$p_{14}$	$p_{13}$	$p_{12}$	$p_{11}$	$p_{10}$	$p_9$	$p_8$
۳/۳	۳/۷۴۱	۳/۴	۳/۳۶۱	۳/۵۲۶	۳/۰۶۸	۳/۵۰۷
$p_{20}$	$p_{19}$	$p_{18}$	$p_{17}$	$p_{16}$	$p_{15}$	
۴	۳/۹۴۴	۳/۷	۴	۳/۴۱	۴	

نتایج حاصل از الگوریتم GA:

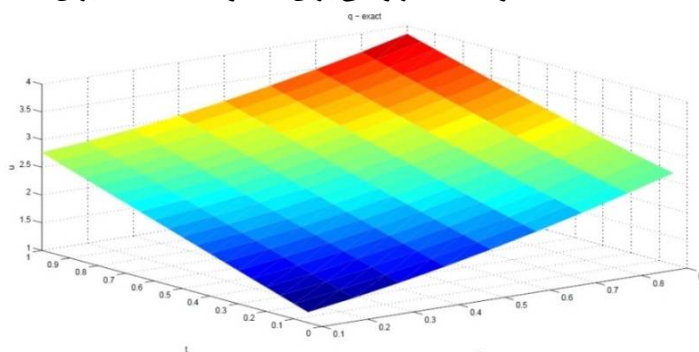
شکل ۱۳. نمودار  $q(t)$  واقعی و  $q(t)$  تقریب زده شده توسط الگوریتم GA برای ۱۰۰ و ۱۰۰۰ تکرار با مقادیر حقیقی، برای ۱۰۰ کروموزوم و  $\beta = ۰/۸۵۶۱$  برای مثال ۲.۴.



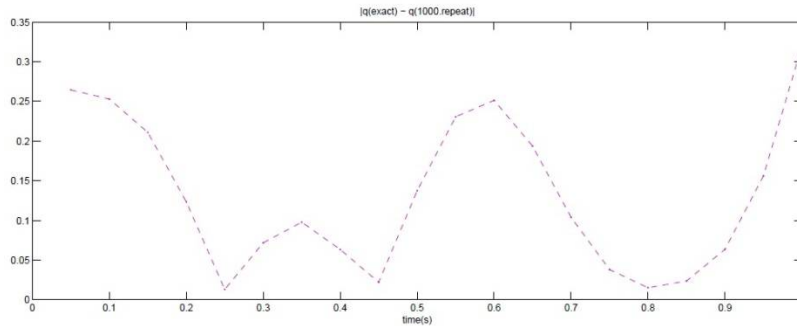
شکل ۱۴. شکل ایجاد شده  $q(t)$  توسط مقادیر بدست آمده از الگوریتم GA برای ۱۰۰۰ تکرار و  $\beta = ۰/۸۵۶۱$  برای مثال ۲.۴.



شکل ۱۵. شکل ایجاد شده توسط مقادیر واقعی برای  $q(t)$  و  $\beta = ۰/۸۵۶۱$  برای مثال ۲.۴.



شکل ۱۶. قدر مطلق تفاضل  $q(t)$  واقعی و  $q(t)$  بدست آمده از الگوریتم GA برای ۱۰۰۰ تکرار برای مثال ۲.۴.



جدول ۷. نتایج اجرای الگوریتم GA با مقادیر حقیقی برای پیدا کردن شرط مجهول  $q(t)$  برای ۱۰۰ و ۱۰۰۰ تکرار برای مثال ۲.۴.

تکرار	بهترین برازندگی	S	Time (s)	$\beta$
۱۰۰	۰/۰۰۸۴	۰/۰۳۷۵	۹/۵۴۶۴	۰/۸۵۶۱
۱۰۰۰	$۳/۰۹۷۷ \times ۱۰^{-۵}$	۰/۰۲۷۹	۱۰۳/۶۵۲۳	۰/۸۵۶۱

جدول ۸. مقادیر  $q(t)$  بدست آمده توسط الگوریتم GA برای ۱۰۰ و ۱۰۰۰ تکرار برای مثال ۲.۴.

تکرار							کروموزوم
$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$	$p_7$	
۲/۷۴۸	۲/۶۶۶	۳/۴۶۶	۳/۰۳۳	۳/۰۷۴	۳/۰۶۳	۲/۰۷۴	۱۰۰
۳/۳۶۹	۳/۴۳۱	۳/۲۹۴	۳/۷۰۸	۳/۲۱۰	۳/۷۷۴	۳/۵۸۸	
۳/۷۳۵	۳/۸۱	۳/۹۵۵	۳/۷۴۱	۳/۶۸۲	۳/۴۴۲		
۳/۱۵۶	۲/۶۹	۳/۲۰۷	۲/۹۲۸	۳/۱۲۰	۳/۲۳۶	۳/۱۲۵	
۳/۳۴۵	۳/۳۶۵	۳/۲۶۷	۳/۶۹۷	۳/۲۷۹	۳/۷۸۸	۳/۵۵۶	۱۰۰۰
۳/۷۶۵	۳/۷۷۸	۳/۸۸۵	۳/۸۶۶	۳/۱۵۵	۳/۷۴۹۷		

ژنتیک کمتر است. بخصوص الگوریتم ازدحام ذرات در این مثال، نسبت به مثال قبل بهتر عمل کرده است. همچنین شکل‌های ۱۰ و ۱۴ بدرتیب مربوط به  $U(x, t)$  بدست آمده با  $q(t)$  تقریب زده شده بوسیله الگوریتم ازدحام ذرات و الگوریتم ژنتیک می‌باشند. نوسان ایجاد شده در این اشکال در محدوده  $x=0.9$  به دلیل خطای اندکی است که در تقریب  $q(t)$  مجهول وجود دارد. شکل‌های ۹ و ۱۳ بترتیب نمودارهای  $q(t)$  واقعی و

در مثال ۲.۴. برای نیز پیدا کردن  $q(t)$  مجهول، الگوریتم‌های ژنتیک و ازدحام ذرات با تکرارهای برابر، اجرا شدند. نتایج بدست آمده از اجرای این الگوریتم‌ها در جداول ۵ و ۷ نشان داده شده‌اند. بررسی این نتایج نشان می‌دهد که در این مثال نیز، الگوریتم ازدحام نتایج بهتری هم از نظر زمانی و هم از نظر دقت، نسبت به الگوریتم ژنتیک ایجاد کرده است. زیرا مقدار خطای کلی ( $S$ ) و زمان محاسبه شده در الگوریتم ازدحام ذرات نسبت به

**مثال ۳.۴.** مساله هدایت گرمایی معکوس زیر را در نظر می‌گیریم.

$$u_t(x, t) = \beta u_{xx}(x, t),$$

$$0 < x < 1, \quad 0 < t < 1$$

$$u(x, 0) = \sin(X) \quad 0 < t \leq 1$$

$$u(0, t) = 0, \quad 0 < t \leq 1$$

$$u(1, t) = q(t), \quad 0 < t \leq 1,$$

که در آن  $\beta$  یک عدد تصادفی در بازه (۰ و ۱) می‌باشد. همچنین شرط اضافی مساله به صورت زیر است.

$$u(0.05, t_i) = s(t_i),$$

$$i = 1, 2, 3, \dots, 20 \quad t_i = i \times 0.05,$$

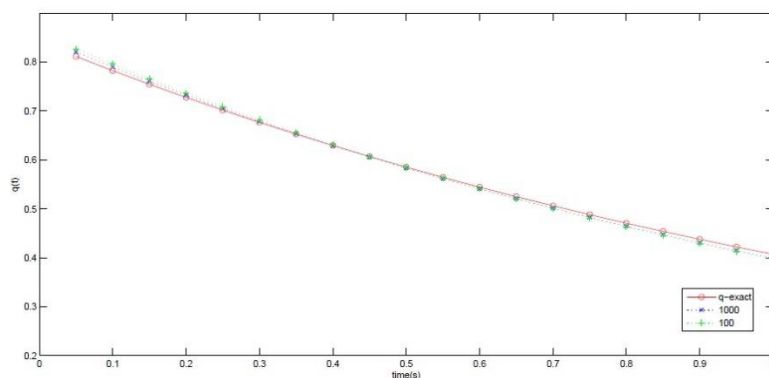
مقدار واقعی  $q(t)$  نیز برابر است با

$$q(t) = e^{-\beta t} \times \sin(1)$$

مجهول تقریب زده شده بوسیله الگوریتم ازدحام ذرات و ژنتیک را نشان می‌دهند. در این اشکال وقتی تعداد تکرار از ۱۰۰ به ۱۰۰۰ تغییر یافته است نمودار  $q(t)$  تقریب زده شده به نمودار  $q(t)$  واقعی نزدیکتر شده است زیرا خطای کلی کمتر شده است. همچنین اشکال ۱۲ و ۱۶ نمودارهای خطای ایجاد شده در تکرار ۱۰۰۰ برای الگوریتم‌های ازدحام ذرات و ژنتیک را نشان می‌دهند. این نمودارها نشان می‌دهند که وقتی تکرار زیاد شده و مقادیر مربوط به  $q(t)$  مجهول به  $q(t)$  واقعی نزدیک می‌شود نقاط متوالی تقرب زده شده برای  $q(t)$  حول نقاط واقعی  $q(t)$  نوسان می‌کنند. در نهایت جداول ۶ و ۸ بترتیب مقادیر تقریب زده شده در نقاط  $x = 0.1, 0.15, 0.20, \dots, 1$  را در تکرارهای ۱۰۰ و ۱۰۰۰ برای الگوریتم‌های ازدحام ذرات و ژنتیک را نشان می‌دهند.

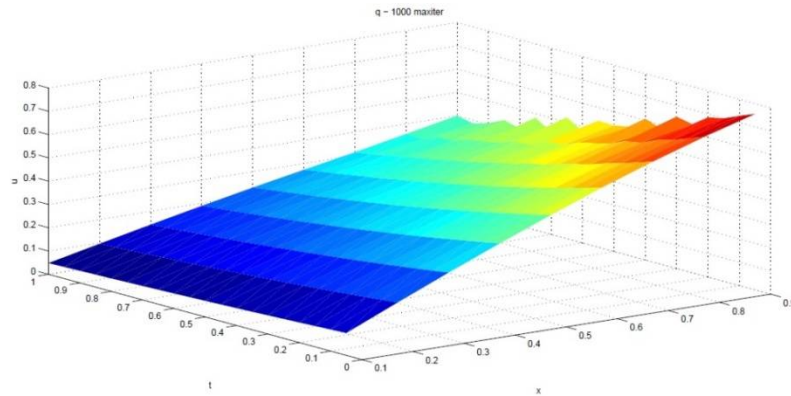
### نتایج حاصل از الگوریتم PSO:

شکل ۱۷. نمودار  $q(t)$  واقعی و  $q(t)$  تقریب زده شده توسط الگوریتم PSO برای ۱۰۰ و ۱۰۰۰ تکرار با مقادیر حقیقی، برای ۱۰۰ ذره و  $\beta = 0.7260$  برای مثال ۳.۴.

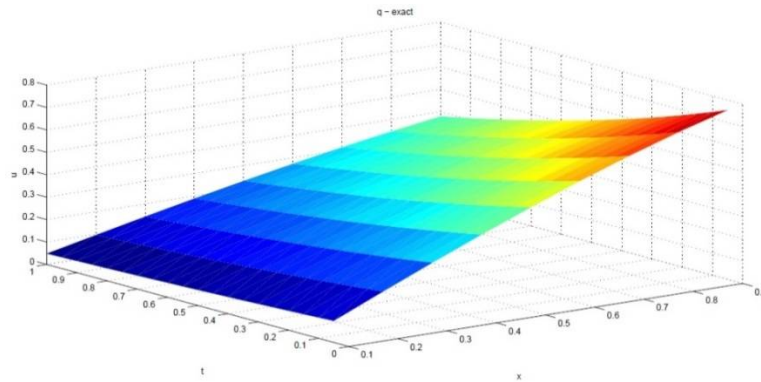


شکل ۱۸. شکل ایجاد شده  $q(t)$  توسط مقادیر بدست آمده از الگوریتم PSO برای ۱۰۰۰ تکرار و  $\beta = 0.7260$  برای

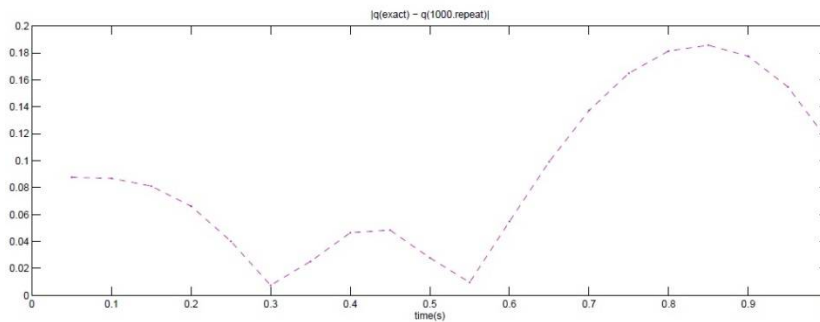
مثال ۳.۴



شکل ۱۹. شکل ایجاد شده توسط مقادیر واقعی برای  $q(t)$  و  $\beta = 0.7260$  برای مثال ۳.۴



شکل ۲۰. قدر مطلق تفاضل  $q(t)$  واقعی و  $q(t)$  بدست آمده از الگوریتم PSO برای ۱۰۰۰ تکرار برای مثال ۳.۴





جدول ۹. نتایج اجرای الگوریتم PSO با مقادیر حقیقی برای پیدا کردن شرط مجهول  $q(t)$  برای ۱۰۰ و ۱۰۰۰ تکرار برای

مثال ۳.۴.

$\beta$	Time (s)	S	بهترین برازندگی	تکرار
۰/۷۲۶۰	۹/۰۶۱۶	۰/۰۱۵۵	$۱/۲۲۹۵ \times ۱۰^{-۵}$	۱۰۰
۰/۷۲۶۰	۸۱/۰۸۱۵	۰/۰۱۰	$۴/۹۴۳۹ \times ۱۰^{-۷}$	۱۰۰۰

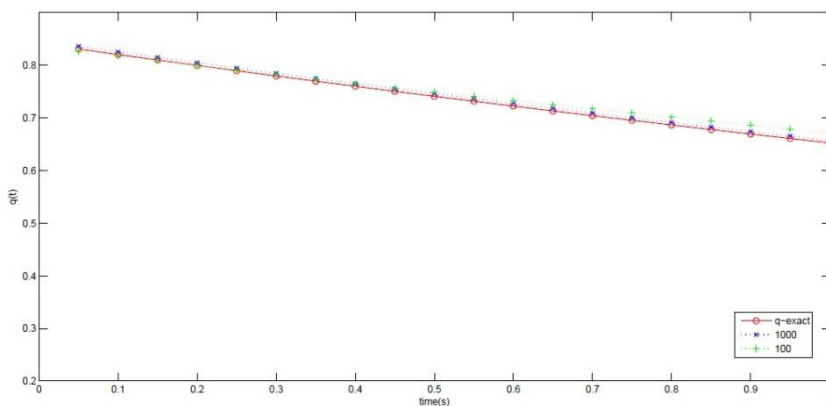
جدول ۱۰. مقادیر  $q(t)$  بدست آمده توسط الگوریتم PSO برای ۱۰۰ و ۱۰۰۰ تکرار برای مثال ۳.۴.

تکرار							کروموزوم										
۱۰۰							$p_7$	$p_6$	$p_5$	$p_4$	$p_3$	$p_2$	$p_1$				
۱۰۰۰							$p_{14}$	$p_{13}$	$p_{12}$	$p_{11}$	$p_{10}$	$p_9$	$p_8$				
۱۰۰۰							$p_{20}$	$p_{19}$	$p_{18}$	$p_{17}$	$p_{16}$	$p_{15}$					
۱۰۰۰							$p_7$	$p_6$	$p_5$	$p_4$	$p_3$	$p_2$	$p_1$				
۱۰۰۰							$p_{14}$	$p_{13}$	$p_{12}$	$p_{11}$	$p_{10}$	$p_9$	$p_8$				
۱۰۰۰							$p_{20}$	$p_{19}$	$p_{18}$	$p_{17}$	$p_{16}$	$p_{15}$					

نتایج حاصل از الگوریتم GA:

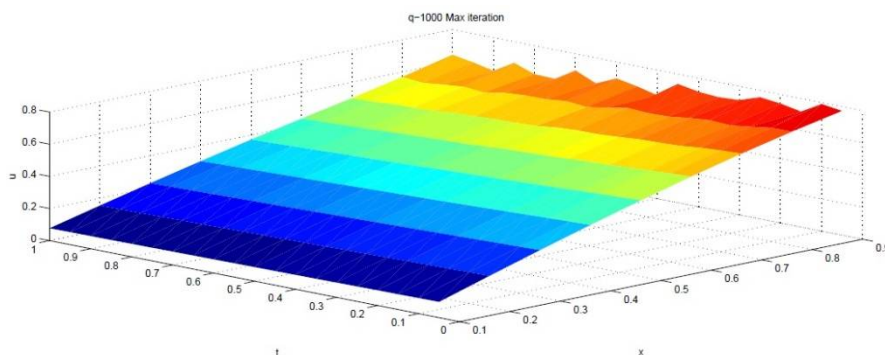
شکل ۲۱. نمودار  $q(t)$  واقعی و  $q(t)$  تقریب زده شده توسط الگوریتم GA برای ۱۰۰ و ۱۰۰۰ تکرار با مقادیر حقیقی، برای

۱۰۰ کروموزوم و  $\beta = ۰/۹۴۸$  برای مثال ۳.۴.

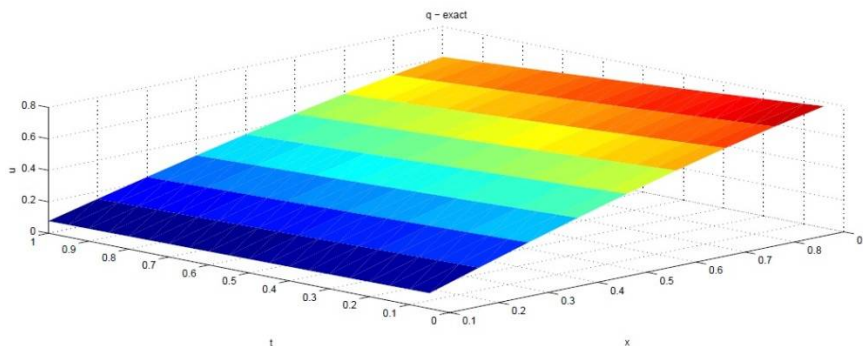


شکل ۲۲. شکل ایجاد شده  $q(t)$  توسط مقادیر بدست آمده از الگوریتم GA برای ۱۰۰۰ تکرار و  $\beta = ۰/۹۴۸۰$  برای

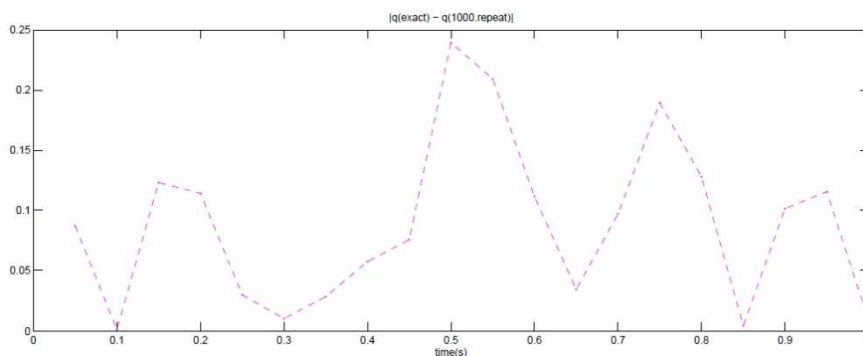
مثال ۳.۴



شکل ۲۳. شکل ایجاد شده توسط مقادیر واقعی برای  $q(t)$  و  $\beta = ۰/۹۴۸۰$  برای مثال ۳.۴



شکل ۲۴. قدر مطلق تفاضل  $q(t)$  واقعی و  $q(t)$  بدست آمده از الگوریتم GA برای ۱۰۰۰ تکرار برای مثال ۳.۴



جدول ۱۱. نتایج اجرای الگوریتم GA با مقادیر حقیقی برای پیدا کردن شرط مجهول  $q(t)$  برای ۱۰۰ و ۱۰۰۰ تکرار برای

مثال ۳.۴

$\beta$	Time (s)	S	بهترین برازندگی	تکرار
۰/۹۴۸۰	۹/۱۳۳۲	۰/۰۵۳	$۲/۹۷۳۰ \times ۱۰^{-۴}$	۱۰۰
۰/۹۴۸۰	۹۴/۸۹۲	۰/۰۲۷۹	$۵/۹۶۰۱ \times ۱۰^{-۵}$	۱۰۰۰

جدول ۱۲. مقادیر  $q(t)$  بدست آمده توسط الگوریتم GA برای ۱۰۰ و ۱۰۰۰ تکرار برای مثال ۳.۴.

تکرار							کروموزوم
$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$	$p_7$	
۰/۹۲۱	۰/۷۵۰	۰/۶۰۵	۰/۸۲۹	۰/۶۴۵	۰/۵۸۲	۰/۶۰۳	۱۰۰
$p_8$	$p_9$	$p_{10}$	$p_{11}$	$p_{12}$	$p_{13}$	$p_{14}$	
۰/۶۲۵	۰/۵۴۳	۰/۴۲۸	۰/۶۲۱	۰/۴۵۰	۰/۳۷۳	۰/۵۴۴	
$p_{15}$	$p_{16}$	$p_{17}$	$p_{18}$	$p_{19}$	$p_{20}$		
۰/۳۸۹	۰/۲۸۸	۰/۶	۰/۰۵۸	۰/۵۱۸	۰/۵۳۶۶		
$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$	$p_7$	
۰/۷۸۳	۰/۷۹۳	۰/۶۷۲	۰/۷۷۲	۰/۶	۰/۶۶۶	۰/۵۹۱	
$p_8$	$p_9$	$p_{10}$	$p_{11}$	$p_{12}$	$p_{13}$	$p_{14}$	
۰/۵۷۲	۰/۵۹۲	۰/۴۳۹	۰/۵۸۱	۰/۴۵۳	۰/۴۰۵	۰/۵۰۹	
$p_{15}$	$p_{16}$	$p_{17}$	$p_{18}$	$p_{19}$	$p_{20}$		
۰/۳۹۲	۰/۲۹۰	۰/۶۰۵	۰/۰۸۴	۰/۵۳۴	۰/۳۵		

در مثال ۳.۴ برای پیدا کردن  $q(t)$  مجهول، الگوریتم‌های ژنتیک و ازدحام ذرات با تکرارهای برابر، اجرا شدند. نتایج بدست آمده از اجرای این الگوریتم‌ها در جداول ۹ و ۱۱ نشان داده شده اند. مقایسه این نتایج نشان می‌دهد که الگوریتم ازدحام نتایج بهتری هم از نظر زمانی و هم از نظر دقت، نسبت به الگوریتم ژنتیک ایجاد کرده است. همچنین شکل‌های ۲۲ و ۱۸ بدرتیب مربوط به  $U(x, t)$  بدست آمده با  $q(t)$  تقریب زده شده بوسیله الگوریتم ازدحام ذرات و الگوریتم ژنتیک می‌باشند. نوسان ایجاد شده در این اشکال در محدوده  $x=0,9$  به دلیل خطای اندکی است که در تقریب  $q(t)$  مجهول وجود دارد. شکل‌های ۱۷ و ۲۱ بترتیب نمودارهای  $q(t)$  واقعی و مجهول تقریب زده شده بوسیله الگوریتم ازدحام ذرات و ژنتیک را نشان می‌دهند. در این اشکال وقتی تعداد تکرار از ۱۰۰ به ۱۰۰۰ تغییر می‌یابد نمودار  $q(t)$  تقریب زده شده به نمودار  $q(t)$  واقعی نزدیکتر می‌شود زیرا خطای کلی کمتر می‌شود. همچنین اشکال ۲۰ و ۲۴ نمودارهای خطای ایجاد شده در تکرار ۱۰۰۰ برای الگوریتم‌های ازدحام ذرات و ژنتیک را نشان می‌دهند. این نمودارها بیان می‌کنند که وقتی تکرار زیاد شده و مقادیر مربوط به  $q(t)$  مجهول به  $q(t)$  واقعی نزدیک می‌شود نقاط متوالی تقرب زده شده برای  $q(t)$  حول نقاط واقعی

در مثال ۳.۴ نوسان می‌کنند. در نهایت جداول ۱۰ و ۱۲ بترتیب مقادیر تقریب زده شده در نقاط  $x = 0.1, 0.15, 0.20, \dots, 1$  برای  $q(t)$  را در تکرارهای ۱۰۰ و ۱۰۰۰ را برای الگوریتم‌های ازدحام ذرات و ژنتیک را نشان می‌دهند.

### ۵- طرح بحث

معمولاً هدف استفاده از الگوریتم‌های تکاملی پیدا کردن یک یا چند بردار بعنوان جواب بهینه یک مساله می‌باشد. معمولاً این بردارها طوری پیدا می‌شوند که تابع هدف را در یک مساله بهینه کنند. در این مقاله تابع هدف مساله، همان خطای کلی حاصل از تقریب بردار مجهول مساله می‌باشد. هر اندازه بردار تقریب زده شده برای  $q(t)$  به بردار واقعی  $q(t)$  نزدیکتر باشد مقدار خطای کلی کاهش می‌یابد. پس هدف اصلی کاهش دادن خطای کلی یا همان بهینه کردن خطای کلی مساله می‌باشد. الگوریتم‌های انبوه ذرات و ژنتیک یکی از قدرتمندترین و اساسی ترین الگوریتم‌های تکاملی می‌باشند و هدف اصلی این مقاله نشان دادن قابلیت این الگوریتم‌ها برای حل مسائل هدایت گرمایی تصادفی معکوس، وقتی که یکی از شرایط مرزی مجهول می‌باشد است. نتایج بدست آمده از مثال های این مقاله نشان می‌دهد که این

- هر چقدر تعداد تکرار بیشتر باشد احتمال پیدا شدن جواب بهتر برای مساله بهتر می‌شود زیرا در مثال‌های این مقاله تکرار ۱۰۰۰ جواب‌های بهتری نسبت به تکرار ۱۰۰ تولید کرد.
- زمان محاسبه شده برای روش‌های مطرح شده در این مقاله زمان قابل قبولی می‌باشد. بنابراین در کاربردهای عملی نیز می‌توانند مورد استفاده قرار گیرند.
- توانایی حل الگوریتم‌های ازدحام ذرات و ژنتیک برای حل این مسائل نشان می‌دهد الگوریتم‌های تکاملی دیگری نیز می‌توانند این نوع مسائل را حل کنند که می‌توانند در تحقیقات آتی مورد بررسی قرار گیرند.
- از آنجایی که موازی سازی الگوریتم‌های مطرح شده در این مقاله امکان‌پذیر است و این الگوریتم‌ها در طبیعت ذاتاً موازی هستند پس بنابراین موازی کردن آنها می‌تواند زمان قابل قبول تری ارائه دهد که می‌تواند زمینه‌ساز تحقیقات آتی در این زمینه باشند.

الگوریتم‌ها برای پیدا کردن بردار مجهول مساله که همان  $q(t)$  می‌باشند خوب عمل کرده‌اند و جواب‌های دقیقی را تولید کرده‌اند. همچنین بررسی‌ها نشان می‌دهد در حالت کلی الگوریتم ازدحام ذرات نسبت به الگوریتم ژنتیک بهتر عمل کرده است. زیرا زمان لازم برای محاسبات و همچنین خطای کلی محاسبه شده، در الگوریتم ازدحام ذرات نسبت به الگوریتم ژنتیک کمتر است. نمودارهای مربوط به  $q(t)$  واقعی و تقریب زده شده نشان می‌دهند که در ابتدای بازه یعنی  $x=0$  مقدار  $q(t)$  واقعی و تقریب زده شده نزدیک بهم می‌باشند زیرا در نزدیکی این نقطه، مقادیر دقیق شرط مرزی که همان  $U(0, t)$  می‌باشد معلوم است ولی با نزدیکتر شده به نقطه  $x=0/9$  دو نمودار اندکی از فاصله می‌گیرند که امر اجتناب ناپذیر می‌باشد زیرا شرط مرزی  $U(1, t)$  مجهول می‌باشد. نتایج کلی بدست آمده برای  $U(x, t)$  در مثال‌های این مقاله بیانگر این مطلب است که خطای حاصل از تقریب  $q(t)$  باعث ایجاد نوسان اندکی در نمودارهای مربوط به  $U(x, t)$  شده است که با افزایش تعداد تکرار این خطا کاهش یافته و نوسانات در بازه کوچکتري اتفاق می‌افتند.

## ۶- نتیجه‌گیری

نتایج بدست آمده توسط الگوریتم‌های ژنتیک و ازدحام ذرات برای حل مسائل هدایت گرمایی تصادفی معکوس به شرح زیر می‌باشد.

- با توجه به اینکه این نوع از مسائل دارای پارامتر تصادفی‌ای می‌باشند، الگوریتم‌های تکاملی که جزو الگوریتم‌های فراابتکاری محسوب می‌شوند و مستقل از مساله هستند گزینه بسیار مناسبی برای حل آنها می‌باشند.
- هر دو الگوریتم ژنتیک و ازدحام ذرات برای حل مسائل هدایت گرمایی تصادفی معکوس، وقتی که یکی از شرایط مرزی مجهول باشد مناسب هستند و می‌توانند یک جواب تقریبی مناسب برای این نوع مسائل تولید کنند. اما در حالت کلی الگوریتم ازدحام ذرات در مسائل مربوط به این مقاله بهتر عمل کرده است.

## فهرست منابع

- [8] S. Foadian, R. Pourgholi, S. H. Tabasi, and J. Damirchi, "The inverse solution of the coupled nonlinear reaction-diffusion equations by the Haar wavelets", *International Journal of Computer Mathematics*, Volume 96, Issues 1, 2019, Pages 105-125.
- [9] S. Foadian, R. Pourgholi, and S. H. Tabasi, "Cubic B-spline method for the solution of an inverse parabolic system", *Applicable Analysis and International Journal*, Volume 97, Issues. 3, 2018, Pages 438-465.
- [10] R. Pourgholi, A. Saeedi, "Applications of cubic B-splines collocation method for solving nonlinear inverse parabolic partial differential equations", *Numerical Methods for Partial Differential Equations*, Volume 33, Issues 1, 2017, Pages 88-104.
- [11] R. Pourgholi, A. Saeedi, "Solving a nonlinear inverse problem of identifying an unknown source term in a reaction-diffusion equation by Adomian Decomposition method", *TWMS Journal of Applied and Engineering Mathematics*, Volume 6, Issues 1, 2016, Pages 150.
- [12] Clerc, M. and Kennedy, J. "The Particle Swarm: Explosion, Stability and Convergence in multi-dimensional complex space." *IEEE Transactions on Evolutionary Computation.*, Vol.20, No. 6, 2002, pp. 58-73.
- [13] Siarry, P., *Metaheuristics for Hard Optimizations*. Springer-Verlag Berlin Heidelberg, 2006.
- [14] Yamian, Peng, Ji Nan, and Zhang Huancheng. "Improved PSO Research for Solving the Inverse Problem of Parabolic Equation." *International Journal of Database Theory and Application*, Vol. 9, No.12, 2016, pp. 173-184.
- [۱] عظیمی عزیز، غلامی شهاب و بیاتی حسن، "بررسی انتقال حرارت هدایت غیرفوری‌ای در شن با استفاده از مدل‌های بر پایه حساب کسری"، *نشریه مدل‌سازی در مهندسی*، سال پانزدهم، شماره ۴۸، ۱۳۹۶، صفحه ۱۵-۲۲.
- [۲] عابدی فرد رامین و سعدالدین سیف‌الله، "مدل کردن جریان سیال مذاب و انتقال حرارت غیرفوری‌ای در جوشکاری سوراخ کلیدی با قوس پلاسما"، *نشریه مدل‌سازی در مهندسی*، سال چهاردهم، شماره ۴۴، ۱۳۹۵، صفحه ۳۵-۴۷.
- [۳] ارمانی طاهر، مغربی محمد جواد و نظری محسن، "تأثیرات پارامتر انتقال حرارت سیال/نانوذره و سیال/جسم جامد بر روی انتقال حرارت جابجایی اجباری نانو سیالات در کانال متخلخل: با شرط عدم تعادل حرارتی"، *نشریه مدل‌سازی در مهندسی*، سال چهاردهم، شماره ۴۴، ۱۳۹۵، صفحه ۱۱-۲۰.
- [۴] گودرزی کوروش و رضانی سید رضا، "مقایسه روش‌های انتقال حرارت معکوس جهت تخمین ضریب انتقال حرارت تماسی"، *نشریه مدل‌سازی در مهندسی*، سال دوازدهم، شماره ۳۷، ۱۳۹۳، صفحه ۲۵-۳۸.
- [۵] کریمی زاد گوهری فاطمه و شاهسوند اکبر، "مقایسه نتایج حاصل از شبکه‌های عصبی MLP و RBF در پیش‌بینی نتایج حاصل از همزمانی پدیده‌های انتقال جرم و انتقال حرارت"، *نشریه مدل‌سازی در مهندسی*، سال یازدهم، شماره ۳۳، ۱۳۹۲، صفحه ۲۷-۴۳.
- [6] Namjoo, M., and A. Mohebbian. "Finite Difference Methods For Random Partial differential Equations", 3rd Conference on Financial Mathematics & Applications. 30, 31 January 2013, Semnan University, Semnan, Iran, L17-M610, pp.1-5.
- [7] Bullheimer, B., Hartel R. and Strauss, C. "A new rank-based version of the ant system: A computational study." *Central European Journal of Operations Research and Economics.*, No. 7, 1999, pp.25-38.

genetic algorithm: Sequential and multi-core parallelization approach.” *Applied Mathematical Modelling*, Vol. 38, No.7, 2014, pp.1948-1958.

[24] F.-B. Liu, A modified genetic algorithm for solving the inverse heat transfer problem of estimating plan heat source, *Int. J. Heat Mass Transfer*, Vol. 51, 2008, pp. 3745-3752.

[25] Mateescu, George Daniel. ”On the application of genetic algorithms to differential equations.” *Romanian Journal of Economic Forecasting* Vol. 7, No. 2, 2006, pp.5-9.

[26] B. M. Kim, Y. B. Kim, C. H. Oh, “A study on the convergence of genetic algorithm”, *Computers & Industrial Engineering*, Volume 33, Issues 3–4, December 1997, Pages 581-588.

[15] Colaço, Marcelo J., Helcio RB Orlande, and George S. Dulikravich. ”Inverse and optimization problems in heat transfer.” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 28, No.1, 2006, pp. 1-24.

[16] Mu, Ai-Qin, De-Xin Cao, and Xiao-Hua Wang. ”A modified particle swarm optimization algorithm.” *Natural Science* Vol.1, No.2, 2009, pp. 151-155.

[17] Olariu, S. and Zomaya, A.Y., *Handbook of Bioinspired Algorithms and Applications*, Taylor Francis Group, LLC Press, 2006.

[18] Qing, A. ”Electromagnetic Inverse Scattering of Multiple Two-Dimensional Perfectly Conducting Objects by the Differential Evolution Strategy.” *IEEE Transactions on Antennas and Propagation*, Vol. 51, No. 6, pp. 1251-1262, 2003.

[19] Hoorfar, A. ”Evolutionary Programming in Electromagnetic Optimization”, *IEEE Transactions on Antennas and Propagation*, Vol. 55, No. 3, pp. 523-537, March 2007.

[20] Glover F. and Kochenberger, G. A. *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003.

[21] J.H. Holland, *Adaptation in Natural and Artificial System*, University of Michigan Press, Ann Arbor, 1975.

[22] W. Qian, M. Li, “Convergence analysis of standard particle swarm optimization algorithm and its improvement”, *Soft Computing*, June 2018, Volume 22, Issue 12, pp 4047–4070.

[23] Pourgholi, Reza, Hassan Dana, and Seyed Hashem Tabasi. ”Solving an inverse heat conduction problem using