مجموعه مقالات
دهمین کنفرانس مهندسی پزشکی ایران

تکنیک جدید برای ایجاد جمعیت اولیه در الگوریتمهای ژنتیکی

محمد حسین صداقی رضا ارجمند

گروه مهندسی پزشکی، دانشکده برق، دانشگاه صنعتی سهند، تبریز، ایران
Email: arjomand_reza@yahoo.com sedaaghi@yahoo.com.

چکیده

این مقاله روش جدیدی را برای ایجاد جمعیت اولیه در الگوریتمهای ژنتیکی به نام Noisy Normal معرفی می‌کند. در این روش، برخلاف روش Quasi random، علاوه بر اینکه از تمام نقاط دامنه (در هر بعد) نماینده‌ای وجود دارد، نقاط انتخابی در اجراهای متفاوت تغییر نیز می‌کنند. با توجه به توزیع مناسب این نقاط، احتمال اینکه الگوریتم برای تعداد جمعیت کم در تله‌های احتمالی بیفتد، پایین می‌آید. روش ارایه شده مشکل توزیع نامناسب Pseudo random و همچنین مشکل عدم تصادفی بودن جمعیت اولیه و توزیع نامناسب برای توابع با ابعاد بالا در روش Quasi random را نیز اصلاح می‌نماید و منجر به پاسخ مطمئن‌تری می‌شود. نتایج آماری بدست آمده این مطالب را تأیید می‌کنند.

واژه‌های کلیدی: الگوریتمهای ژنتیکی، توزیع جمعیت روی دامنه، جمعیت اولیه Noisy Normal.

۱- مقدمه

الگوریتمهای ژنتیکی (GAs) یکی از جدیدترین دستاوردها در زمینه بهینه‌سازی توابع می‌باشند که با استفاده از یک سری اپراتورهای تعریف شده در GA و مبتنی بر انتخاب تصادفی، که برپایه نظریه تکامل تدریجی داروین می‌باشد، به صورت global قادر به کشف نقاط بهینه هستند [۱]. GAs امروزه در بیشتر رشته‌های علوم (از طراحی فیلتر گرفته تا تصویربرداری پزشکی، تعیین ساختارهای غیرخطی دینامیکی، سیستمهای پیونددهنده و فرآیندهای مولکولی و ...) در صنعت و حتی بازارهای بورس، کاربردهای فراوانی دارد و روزبه‌روز کاربردهای جدیدتری برای آن پیدا می‌شود [۱،۲]. شمای کلی این الگوریتمها به شرح زیر است:

۱) این الگوریتمها ابتدا یک سری نقاط را به شکل تصادفی و به تعداد مشخص از دامنه تابع انتخاب می‌کنند (مرحله initialization). این نقاط معادل ژن های موجود در طبیعت میباشند.

۲) سپس با توجه به ارزش (fitness)، تعدادی از آن‌ها به شکل تصادفی برای ادامه کار انتخاب می‌گردند. بدیهی است احتمال بقای ژن‌های قوی‌تر بیشتر خواهد بود.

۳) مرحله بعدی، crossover است (که یکی از اپراتورهای GAs برای بهینه کردن توابع می‌باشد و از mating در طبیعت الهام گرفته شده است). در این مرحله، برخی از نقاط به صورت تصادفی انتخاب گشته و عمل crossover را انجام می‌دهند.

٤) پس از این تعدادی از جمعیت به صورت تصادفی برای عمل mutation انتخاب می‌شوند.

مرحله ۲ تا ٤ به تعداد معینی به صورت تکراری انجام می‌پذیرد و در پایان بهترین نقطه (نقاط) اعلام می‌گردد(ند).

پس در این عملیات نحوه انتخاب جمعیت اولیه حایز اهمیت می‌باشد.

در این زمینه دو روش عمده بکار می‌روند که عبارتند از Pseudo Random Sequence (PRS) و Quasi Random Sequence (QRS). البته هر یک از آن‌ها دارای مزایا و معایبی هستند که در بخش ۲ به شرح آن‌ها خواهیم پرداخت. سپس در بخش ۳ روش Noisy Normal Sequence (NNS) را مورد مطالعه قرار می‌دهیم. در بخش ٤ این سه روش را بر روی سه تابع مشهور برای بررسی آماری امتحان خواهیم نمود. در بخش ۵ به بررسی نتایج آماری خواهیم پرداخت.

۲- بررسی دو روش موجود

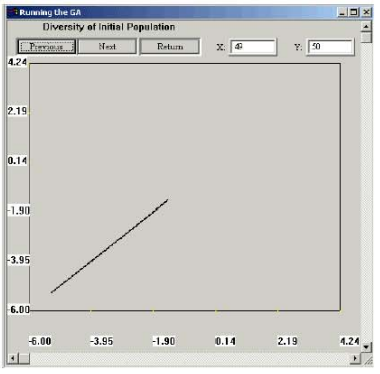
در روش اول، که به نام PRS معروف است، همان‌گونه که از نام آن برمی‌آید، جمعیت اولیه کاملاً به شکل تصادفی از دامنه مشخص شده انتخاب می‌شوند. با وجود اینکه این روش کارایی فراوانی دارد،

اما بنا به تصادفی بودنش اشکالاتی بر آن وارد است. نظر به اینکه روش PRS توزیع مشخصی ندارد، احتمال اینکه همه دامنه را تحت پوشش قرار ندهد (یا حداقل به شکل نسبتاً یکنواخت بپوشاند)، بسیار زیاد است و همین مسأله احتمال در تله افتادن را بالا می‌برد، که به عنوان یک نقیصه مورد توجه است.

اما در روش QRS [۳] که اساس آن انتخاب اعداد طبیعی به ترتیب صعودی به عنوان مبنا می‌باشد، به ترتیب برای دامنه در بعد اول، دوم، سوم و ...، اعداد طبیعی ۲، ۳، ۵ و ... به شکلی که در زیر توضیح داده می‌شود، معیار کار قرار می‌گیرند. روش کار برای بعد اول و استفاده از نخستین عدد طبیعی، یعنی ۲، در زیر نشان داده شده است.

$$X_i = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1\times 2^{-1}} & x_1 = \mathbf{0.5} \\ \mathbf{2} & \mathbf{10} & \mathbf{01} & \mathbf{0\times 2^{-1}+1\times 2^{-2}} & x_2 = \mathbf{0.25} \\ \mathbf{3} & \mathbf{11} & \mathbf{11} & \mathbf{1\times 2^{-1}+1\times 2^{-2}} & x_3 = \mathbf{0.75} \\ \mathbf{4} & \mathbf{100} & \mathbf{001} & \mathbf{0\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}} & x_4 = \mathbf{0.125} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \end{bmatrix}$$

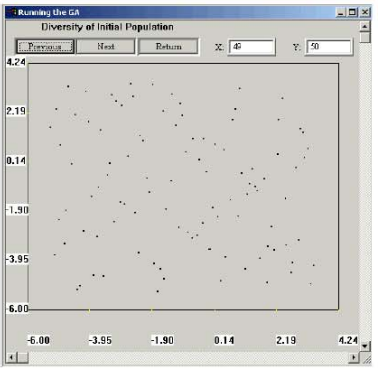
این روش قابل تعمیم برای اعداد طبیعی دیگر در ابعاد متفاوت می‌باشد. ملاحظه می‌گردد که هرچه ابعاد بالا می‌رود عدد طبیعی پایه نیز بزرگتر می‌گردد. بنابراین برای اینکه در توابع با ابعاد بزگ توزیع نسبتاً مناسبی داشته باشیم باید Pop_size (population size): تعداد نقاط انتخابی برای هر بعد) بسیار بزرگی را انتخاب کنیم که این نیاز به حجم عملیاتی و زمان بسیار بالایی خواهد داشت و در غیر این صورت جمعیت دارای توزیع خطی که در اکثر موارد هم کل دامنه را پوشش نمی‌دهد، خواهد بود.



شکل ۱: توزیع جمعیت اولیه بعد ٤٩ نسبت به ٥٠ در روش QRS

همانطوریکه در شکل ۱ نشان داده شده است، توزیع جمعیت در روش QRS در ابعاد بالا کاملاً خطی بوده و بخش کاملاً محدودی از دامنه تحت پوشش قرار می‌گیرد. این مسأله نسبت مستقیمی با بعد تابع و Pop_size انتخاب شده دارد؛ مثلاً برای یک تابع ٥٠ بعدی با ٥٠= Pop_size این مشکل از بعد ٢٠ به بالا مشهود است. این مسأله باعث می‌گردد الگوریتم در ابعاد بالا به تله بیفتد.

در حالیکه، همانگونه که در شکل ۲ دیده می‌شود، روش NNS توزیع بسیار مناسب‌تری دارد.



شکل۲: توزیع جمعیت اولیه بعد ٤٩ نسبت به ٥٠ در روش NNS

علاوه بر آن، اعداد تولید شده در روش QRS در اجراهای متفاوت همواره یکسان خواهند بود. به فرض اگر یک تابع تک بعدی داشته باشیم، اعدادی در مبنای ۲ تولید می‌گردند. برای دامنه بین صفر و یک همواره اعداد ۰/۵، ۰/۲۵، ۰/۷۵ و ... تولید خواهد گردید و برای دامنه‌های دیگر به صورت ضریبی از این اعداد تولید می‌شوند. مثلاً اگر دامنه بین صفر و ده باشد اعداد تولیدی به ترتیب ۵، ۲/۵ ، ۷/۵، ۱/۲۵ و ... (همه اعداد در ده ضرب شده‌اند) خواهند بود. همین مشابه بودن جمعیت اولیه برای دامنه‌های یکسان (با وجود متفاوت بودن توابع) می‌تواند به عنوان یک نقطه کور الگوریتم فوق باشد. به عنوان مثال اگر دامنه از صفر تا یک باشد و Pop_size =۳ و تابعی داشته باشیم که متناوب با دوره تناوب ۰/۲۵ باشد و در ۰/۲۵، ۰/۵ و ۰/۷۵ دارای مینیمم موضعی باشد احتمال در تله افتادن بسیار بالا می‌رود. البته در واقعیت ما به ندرت (تقریباً هرگز) با چنین توابعی و با این تعداد Pop_size مواجه می‌شویم اما همچنان اشکال وارد به قوت خود باقیست. همچنین این نوع انتخاب جمعیت اولیه باعث می‌شود که تصادفی بودن این جمعیت از بین برود که چندان با اصل الگوریتمهای ژنتیکی همخوانی ندارد.

۳- روش ارائه شده

با توجه به معایب دو روش فوق (عدم توزیع یکنواخت و جمعیت اولیه غیر تصادفی و مشابه) به دنبال روشی هستیم که علاوه بر تصادفی بودن توریع یکنواختی نیز داشته باشد. الگوریتم ایجاد جمعیت اولیه در روش NNS بدین ترتیب است که ابتدا مینیمم (Domain_Min) و ماکزیمم (Domain_Max) دامنه مشخص می‌شود سپس گستره دامنه (Range) آن محاسبه می‌گردد: Range = Domain_Max - Domain_Min. سپس با تقسیم دامنه بر Pop_size گام (Step) بدست می‌آید: Step= Range ÷ Pop_size. حال عدد تصادفی $\mathbf{0 \le \alpha \le 1}$ تولید می‌گردد با ضرب این عدد در گام، عدد تصادفی (Rand_num) بدست می‌آید: $Rand\_num_1 = \alpha \times Step$ با اضافه کردن $Rand\_num_1$ روی مینیمم دامنه اولین عضو حاصل می‌گردد. $x_1 = Rand\_num_1 + Domain\_Min$ برای بدست آوردن عضو دوم مینیمم دامنه با گام جمع می‌شود و دوباره یک عدد تصادفی ($Rand\_num_2$) به روش فوق ایجاد و با آن‌ها جمع می‌گردد: $x_2 = Domain\_Min + Step + Rand\_num_2$ و این کار در هر بعد و به تعداد Pop_size تکرار می‌شود. حاصل یک سری اعداد تصادفی با توزیع یکنواخت مناسب تضمین شده روی هر بعد دامنه است.

٤- بررسی آماری

سه روش فوق (PRS,QRS,NNS) را برای مینیمم کردن چهار تابع زیر بکار می‌بریم: این توابع برای امتحان کارایی GAs به عنوان chmark ben بکار می‌روند [٤]:

$$f(x_1,x_2)=100(x_1^2-x_2^2)^2+(1-x_1)^2 \qquad ,-6\le x_i\le 6$$

$$f(x_1,x_2)=x_1^2+2x_2^2-0.3Cos(3\pi x_1)-0.4Cos(4\pi x_2)+0.7 \qquad ,-1\le x_1,x_2\le 1$$

$$f(x_i)=\sum_{i=1}^{10}\frac{(x_i)^2}{4000}-\prod_{i=1}^{10}Cos\frac{x_i}{\sqrt{i}}+1,-600\le x_i\le 600 \qquad ,i=1,...,10$$

$$f(x_i)=50A+\sum_{i=1}^{50}(x_i^2-ACos(2\pi x_i)),-5.12\le x_i\le 5.12 \qquad ,i=1,...,50$$

هر کدام از روشها برای ٥٠ بار روی هر یک از توابع اجرا شده و نتایج آماری در جدول ١ فهرست شده‌اند. در این جدول ستون اول شماره تابع، ستون دوم روشهای سه گانه، ستون سوم میانگین اجراها، ستون ٤ بهترین جواب در بین ٥٠ اجرا، ستون ٥ تعداد دفعاتی است که الگوریتم در تله افتاده و ستون آخر بیانگر تعداد دفعاتی است که جواب بدست آمده از یک حد مشخصی کمتر می‌باشد (که برای تابع ١ عدد ٠/٠٠١، تابع ٢ عدد 10^{-7}، تابع ٣ عدد ١ و برای تابع ٤ عدد ٦ تعیین شده‌اند) نتایج آماری براساس استفاده از بسته نرم‌افزاری SGA، که در دانشگاه سهند و با استفاده از Visual C++5 توسط نویسندگان مقاله طراحی شده است، حاصل گشته‌اند.

جدول ١: نتایج آماری بر روی توابع

تابع	روش	میانگین	بهترین	در تله افتادن	موفقیت
١	PRS	0.03537	0.000969154	6	1
	QRS	0.0463519	0.00127396	1	0
	NNS	0.0359666	0.000610217	2	6
٢	PRS	1.461656×10^{-6}	1.1291×10^{-9}	3	15
	QRS	1.484451×10^{-6}	1.19725×10^{-8}	4	10
	NNS	1.187774×10^{-6}	1.20886×10^{-10}	3	11
٣	PRS	1.281554	0.409928	5	8
	QRS	1.256558	0.735022	4	7
	NNS	1.155243	0.405991	1	20
٤	PRS	8.047313	5.46099	7	2
	QRS	19.007548	12.8786	50	0
	NNS	7.351633	4.79644	4	6

۵- نتیجه‌گیری

روش PRS چون کاملاً تصادفی است قابل اطمینان نمی‌باشد و در بسیاری از موارد در تله‌های موضعی (local) می‌افتد و برای توابع با ابعاد بالا این احتمال بیشتر می‌گردد. روش QRS بخاطر الگوریتم تولیدش در توابع با ابعاد بالا در تله می‌افتد و کارایی لازم را ندارد. روش NNS در تئوری مشکل توزیع جمعیت اولیه بر روی دامنه‌های مشخص شده در هر بعد را تا حدود بسیار زیادی حل کرده است و در عمل نیز با توجه به نتایج بدست آمده ملاحظه می‌گردد که در اکثر موارد منجر به نتایج بهتر و مطمئن‌تری می‌شوند.

مراجع

[1] Goldberg. D. "Genetic Algorithms in search optimization and machine learning", 1989, Addison Wesly.

[2] Caponetto, R., Fortuna, L., Graziani, S., Xibilia, M. G., "Genetic Algorithms and applications in system engineering: a survey." Transaction on Inst. Of Measurement and Control, 15 (3), 1993, 143-156.

[3] Cao, Y. J., Wu, Q. H., "Study of initial population in evolutionary programming" Technical report, Dept. of Elec. Eng., Liverpool University, 1998.

[4] Dekkers, A., Aarts, E., "Global optimization and simulated annealing." Mathematical Programming, 50, 1991, 367-393.