

مقایسه‌ای بین روشهای محلی، سراسری و ترکیبی و ارائه یک روش جدید برای آموزش بهینه مدل پنهان مارکف

محمد رضا میبیدی
استاد

جهانشاه کبودیان
دانشجوی دکترا

محمد مهدی همایون پور
استادیار

دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر

چکیده

یکی از ابزارهای بسیار قدرتمند در پردازش فرآیندهای اتفاقی و دنباله‌های تصادفی مدل پنهان مارکف یا HMM¹ می‌باشد. مشهورترین روش آموزش مدل پنهان مارکف روش بام - ولش² است که یک روش آموزش (جستجوی محلی³ بوده و در دام بهینه‌های محلی گرفتار می‌آید. در این تحقیق از روشهای جستجوی سراسری مبتنی بر سرد کردن فلزات (SA⁴) و روش مبتنی بر اتوماتون یادگیر تقویتی با مقدار عمل پیوسته (CARLA⁵) استفاده کردیم. آزمایشها نشان می‌دهند که روشهای پیشرفته مبتنی بر SA از روش CARLA بهتر هستند، البته روش CARLA نسبت به نسخه استاندارد SA یعنی BA⁶ بهتر عمل می‌کند. همچنین یک روش SA حافظه دار و کاملاً جدید بنام MiGSA⁷ نیز ارائه داده‌ایم که دارای راندمان بالاتری نسبت به نسخه‌های استاندارد و پیشرفته از SAهای بدون حافظه است. آزمایشها نشان داد که کارایی متوسط الگوریتم BW بیش از کارایی متوسط روشهای سراسری در تعداد تکرارهای نه خیلی زیاد (۱۰۰۰۰ تکرار) است و این به دلیل قدرت بالای روش BW در تنظیم دقیق پارامترها، سرعت آن و پایه ریاضی مستحکم آن می‌باشد. برای رسیدن به بهینه سراسری در تعداد تکرارهای نه خیلی زیاد از یک روش ترکیبی⁸ استفاده کردیم که هم از روش جستجوی سراسری مبتنی بر SA و هم از روش جستجوی محلی BW بهره می‌گیرد و دارای کارایی بالاتری نسبت به روشهای سراسری و روش BW به تنهایی می‌باشد. با استفاده از روش ترکیبی فوق الذکر توانسته‌ایم به آموزش بهینه مدل پنهان مارکف بسیار نزدیک شویم.

کلمات کلیدی

مدل پنهان مارکف، روش بام - ولش، جستجوی محلی، جستجوی سراسری، جستجوی ترکیبی، سرد کردن فلزات، اتوماتون یادگیر تقویتی با مقدار عمل پیوسته.

A Comparison of Local, Global and Hybrid Search Techniques and Presenting A New Method for Optimal Training of Hidden Markov Models

J. Kabudian
PhD Student

M.R. Meybodi
Professor

M. M. Homayounpour
Assistant Professor

Department of Computer Engineering and Information Technology,
AmirKabir University of Technology

Abstract

One of the most powerful tools for modeling stochastic processes and sequences is Hidden Markov Model (HMM). The most popular method for training HMMs is Baum-Welch (BW) Re-Estimation which is a local search method and suffers from local minima problem. In this paper, we propose methods based on Continuous Action Reinforcement Learning Automata (CARLA) and Simulated Annealing (SA) based optimization techniques for training Hidden Markov Model. Experimental results show that methods based on fast versions of SA, outperform CARLA-based methods, but CARLA-based methods have higher performance compared to methods based on standard and slow version of SA (i.e. Boltzmann Annealing). Also, we have proposed a new SA-based global search technique with memory called MiGSA (Mixture Generating function SA) which outperforms memoryless standard and fast versions of SA. Experiments show that in a moderate number of iterations (e.g. 10,000), average performance of BW method is higher than that of global search-based methods. This is due to the strength of BW method in fine tuning of parameters, its speed and its tight mathematical basis. For attaining global optimum of training in a moderate number of iterations, we also propose a hybrid technique which benefits from both SA-based global search and BW-based local search. This hybrid technique has a higher performance compared to both SA-based global search and BW-based local search. Using this hybrid technique, we have attained global optimum of HMM training problem.

Keywords

Hidden Markov Model, Baum-Welch Method, Local Search, Global Search, Hybrid Search, Simulated Annealing, Continuous Action Reinforcement Learning Automata.

مقدمه

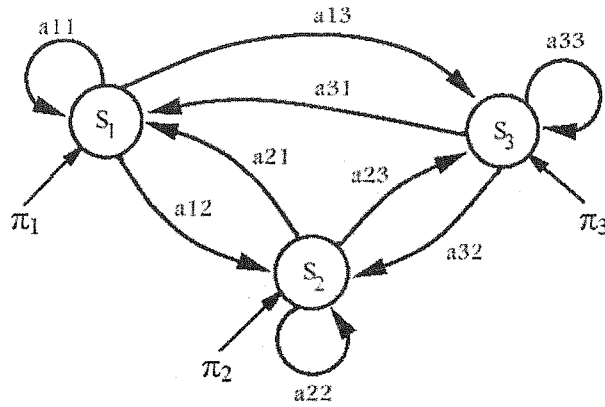
مدل پنهان مارکف یکی از ابزارهای بسیار قوی در پردازش فرآیندهای اتفاقی و دنباله‌های تصادفی است. مشهورترین و پرکاربردترین روش آموزش مدل پنهان مارکف روش بام - ولش است که در واقع یک روش جستجوی محلی است و مانند روشهای دیگر جستجوی محلی در دام بهینه‌های محلی گرفتار می‌آید. برای رفع این ایراد، از روشهای جستجو و بهینه‌سازی سراسری برای آموزش بهینه مدل مارکف استفاده شده است که می‌توانند به راحتی از بهینه‌های محلی بگریزند، ولی این روشها در تنظیم دقیق پارامترهای مدل مارکف و داشتن دقت بسیار بالا، ضعیف هستند و کند می‌باشند. به همین دلیل روشهای ترکیبی برای جستجوی سراسری پیشنهاد شده‌اند که هم از قابلیت روشهای سراسری و هم از قابلیت روشهای محلی توأمأ استفاده می‌کنند. هدف از این کار تحقیقی، مقایسه روشهای محلی، روشهای سراسری و روشهای ترکیبی برای آموزش مدل پنهان مارکف می‌باشد.

در بخشهای بعدی به معرفی مدل پنهان مارکف پیوسته و روش مشهور BW برای آموزش آن خواهیم پرداخت. پس از آن روشهای جستجوی سراسری مبتنی بر SA و اتوماتون یادگیر را به تشریح معرفی خواهیم کرد و سپس روش جستجوی ترکیبی را توضیح خواهیم داد. در پایان هم به ارائه نتایج آزمایشهای انجام شده می‌پردازیم.

۱- مدل پنهان مارکف (HMM)

یکی از ابزارهای بسیار قدرتمند در مدل کردن فرآیندهای اتفاقی و دنباله‌های تصادفی، مدل پنهان مارکف می‌باشد. مدل پنهان مارکف، خود دارای دو نوع پیوسته و گسسته است که نوع گسسته آن در حال حاضر برای مشاهدات پیوسته به ندرت استفاده می‌گردد. نمای یک مدل پنهان مارکف با ۳ حالت در شکل (۱) نشان داده شده است [1].





شکل (۱) یک مدل پنهان مارکوف سه حالتی.

همانطور که مشاهده می‌شود مدل مارکوف دارای تعدادی حالت (N) است که بطور احتمالی و با احتمال گذر بین حالات، تغییر حالت می‌دهد. فرض کنید دنباله‌ای از مشاهدات را که طول دنباله T می‌باشد، توسط این مدل می‌خواهیم تولید کنیم. دنباله مشاهدات چنین تعریف می‌گردد:

$$O = \{o_1, o_2, \dots, o_t, \dots, o_T\} \quad (1)$$

هر کدام از بردارهای مشاهده o_t یک بردار D بعدی می‌باشد که t اندیس زمان است. مدل پنهان مارکوف با N حالت دنباله مشاهدات را چنین تولید می‌کند:

- ۱- با شروع زمان و بازای $t=1$ با احتمال π_i در یکی از حالت‌های مدل (مثلاً i) قرار می‌گیرد.
- ۲- در این حالت یک بردار تصادفی o_1 را با توجه به تابع چگالی احتمال $b_i(o_1)$ تولید می‌کند.
- ۳- مراحل ۴ و ۵ و ۶ تا زمانی که $t < T$ است، تکرار می‌شود و گرنه به مرحله ۷ می‌رویم.
- ۴- $t = t + 1$
- ۵- اگر حالت مدل در زمان $t-1$ برابر i بوده است، مدل در زمان t با احتمال a_{ij} به حالت j زام تغییر حالت می‌دهد.
- ۶- اگر حالت مدل در زمان t برابر i باشد، در این حالت بردار مشاهده o_t را با احتمال $b_i(o_t)$ تولید می‌نماید.

۷- پایان

اگر دنباله حالتی که سیستم در طی زمان طی می‌کند، Q باشد، داریم:

$$Q = \{q_1, q_2, \dots, q_t, \dots, q_T\} \quad (2)$$

q_t حالت سیستم در زمان t است. مدل مارکوف λ را با پارامترهای π ، A، و b() می‌شناسیم و احتمال تولید رشته مشاهدات O توسط مدل مارکوف λ چنین محاسبه می‌گردد:

$$P(O|\lambda) = \sum_{\text{all } q_1, q_2, \dots} \left[\pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^T a_{q_{t-1}q_t} b_{q_t}(o_t) \right] \quad (3)$$

$$P(O|\lambda) = \sum_{\text{all } Q} P(O, Q|\lambda) = \sum_{\text{all } Q} P(O|Q, \lambda) P(Q|\lambda) \quad (4)$$

تابع چگالی احتمال در حالت i که با $b_i(o)$ نشان داده می‌شود، در مدل پنهان مارکوف پیوسته معمولاً یک مخلوط از توابع

گاوسی یا نرمال چند بعدی به شکل زیر است:

$$b_i(o) = \sum_{m=1}^M c_{im} N(o, \mu_{im}, C_{im}) \quad (5)$$

منظور از $N(o, \mu, C)$ یک تابع چگالی احتمال گاوسی D بعدی با بردار میانگین μ و ماتریس کوواریانس C به شکل زیر است:

$$N(o, \mu, C) = \frac{1}{\sqrt{(2\pi)^D |\det(C)|}} \cdot \exp\left(-\frac{1}{2}(o - \mu)^T C^{-1}(o - \mu)\right) \quad (6)$$

فرض بر این است که مشاهدات O_i هر کدام یک بردار ستونی $D \times 1$ باشند. واضح است که برای برقراری شرط یک شدن سطح زیر منحنی تابع چگالی احتمال باید داشته باشیم:

$$\sum_{m=1}^M c_{im} = 1 \quad (7)$$

M تعداد توابع گاوسی در یک مخلوط گاوسی در هر حالت است. همچنین باید داشته باشیم:

$$\sum_{i=1}^N \pi_i = 1 \quad \text{و} \quad \sum_{j=1}^N a_{ij} = 1 \quad (8)$$

C^{-1} معکوس ماتریس C و $|\det(C)|$ قدرمطلق دترمینان ماتریس C می باشد. مدل پنهان مارکف λ با پارامترهای $b(), A, \pi$ چنین نمایش داده میشود:

$$\lambda = (\pi, A, b()) \quad (9)$$

که منظور از $b()$ در واقع وزنه‌های توابع گاوسی (c_{im}) ، بردارهای میانگین (μ_{im}) و ماتریس‌های کوواریانس (C_{im}) است.

۲- آموزش مدل پنهان مارکف با الگوریتم BW

در بکارگیری مدل پنهان مارکف، یک یا چند نمونه از دنباله‌های مشاهدات توسط یک مدل ناشناخته λ تولید شده است و هدف پیدا کردن پارامترهای مدل λ یعنی پارامترهای (A, π, b) است و نتیجتاً مدل پنهان مارکف λ مشخص خواهد شد. مشهورترین و پراستفاده‌ترین روش برای آموزش مدل پنهان مارکف، الگوریتم BW است که بر روش تخمین آماری ML^9 استوار است و در واقع از الگوریتم بیشینه کردن امید ریاضی یا EM^{10} بدست می‌آید. الگوریتم BW از روش EM استاندارد نشأت می‌گیرد و روش EM با همه قدرت و توانایی و پایه ریاضی مستحکم نسبت به خیلی از روشهای آموزش دیگر، دارای ضعفهایی است که باعث شده نسخه‌های پیشرفته‌تر آن نیز در سالهای اخیر پیشنهاد شود [2,3]. هدف الگوریتم BW پیدا کردن مدل λ^* است، طوری که احتمال تولید رشته مشاهدات O توسط این مدل بیشینه باشد:

$$\lambda^* = \arg \max_{\lambda} P(O|\lambda) \quad (10)$$

روش بام - ولش به این صورت عمل می‌کند که ابتدائاً یک حدس اولیه λ_0 بصورت اتفاقی و یا با استفاده از روشهای ابتکاری برای مدل تولید می‌شود. آنگاه در هر مرحله، یک تخمین از پارامترهای مدل در مرحله قبل و با توجه به رشته مشاهدات بدست می‌آید و این تکرار و تخمین آنقدر ادامه می‌یابد تا مدل همگرا شود. الگوریتم بام - ولش بصورت زیر است:

$$\text{Initiate}(\lambda_0), n = 0-1$$

$$n = n + 1-2$$

$$\lambda_n = \text{ReEstimate}(\lambda_{n-1})-3$$

$$4- \text{ اگر } \varepsilon < \left| \frac{P(O|\lambda_n) - P(O|\lambda_{n-1})}{P(O|\lambda_{n-1})} \right| \text{ آنگاه الگوریتم خاتمه می‌یابد و گر نه به قدم 2 خواهیم رفت.}$$

روش تخمین BW از دید ریاضی همیشه تضمین می‌کند که $P(O|\lambda_n) \geq P(O|\lambda_{n-1})$ ، یعنی در هر تکرار، آموزش سیستم رو به بهبود است و این بهبود سیستم ادامه می‌یابد تا وقتی که به نقطه همگرایی یا ماکزیمم محلی یا بهینه محلی تابع $P(O|\lambda)$ بر حسب λ برسیم و در اینجا الگوریتم متوقف خواهد شد و در دام بهینه محلی خواهد افتاد. در این بهینه محلی خواهیم داشت:

$$\frac{\partial P(O|\lambda)}{\partial \lambda} = 0 \quad (11)$$

الگوریتم BW نیز مانند روشهایی مثل حرکت در عکس راستای گرادیان¹¹ GD (انتشار خطا به عقب در شبکه‌های عصبی MLP) یک روش محلی است، ولی برخلاف روشهایی مثل GD تضمین می‌کند که در هر تکرار، راندمان سیستم بهبود می‌یابد و حرکت آن بسیار حساب شده تر و سریعتر است، ولی در GD ممکن است خطای سیستم در یک تکرار نسبت به تکرار قبل بیشتر شود. نکته بسیار مهمی که در مورد الگوریتم BW وجود دارد این است که بر حسب اینکه نقطه شروع اولیه چه باشد، این الگوریتم به بهینه‌های محلی مختلف همگرا خواهد شد و همانطور که خواهیم دید (در بخش آزمایشها)، اگر یک نقطه شروع اولیه A دارای $P(O|\lambda)$ بیشتری نسبت به نقطه شروع اولیه دیگر B باشد، لزومی ندارد که با شروع از نقطه A و اجرای الگوریتم BW به نقطه نهایی بهتری نسبت به شروع از نقطه B برسیم. به عبارت دیگر ممکن است یک نقطه شروع اولیه B نسبت به نقطه شروع اولیه A دارای $P(O|\lambda)$ کمتری باشد (بدتر باشد)، ولی الگوریتم BW با شروع از نقطه اولیه B به بهینه محلی نهایی بهتری نسبت به شروع از نقطه اولیه A برسد. در نظر گرفتن این نکته تأثیر بسیار زیادی در انتخاب نوع الگوریتم خواهد داشت. برای فرار از به دام افتادن در بهینه‌های محلی باید از روشهای جستجوی سراسری یا بهینه سازی سراسری استفاده کرد. برای جلوگیری از اطاله کلام از جزئیات الگوریتم BW خودداری خواهیم کرد و مرجع بسیار خوب [1] را پیشنهاد می‌کنیم.

۳- روشهای جستجو و بهینه‌سازی سراسری

برای حل مسائل بصورت نزدیک به بهینه سراسری^{۱۲} از روشهای جستجوی سراسری^{۱۳} یا بهینه‌سازی سراسری^{۱۴} استفاده می‌شود. از جمله این روشها میتوان به روش سرد کردن فلزات (SA)، روش الگوریتمهای ژنتیکی (GA)¹⁵ [4]، روش استراتژی تکاملی (ES)¹⁶ [4] و روش برنامه‌ریزی تکاملی (EP)¹⁷ [4] اشاره کرد. برای بهینه‌سازی سراسری روشهای دیگری نیز وجود دارد [5]. به تازگی نیز یک روش اتوماتون یادگیر تقویتی با مقدار عمل پیوسته (CARLA) برای بهینه‌سازی سراسری در محیطهای پیوسته ابداع شده است [6,7]. در برخی از این روشها، مثل روش SA رسیدن به بهینه سراسری از دید آماری تضمین شده است، ولی در برخی دیگر همچون GA، ES و EP رسیدن به بهینه سراسری تضمین نشده است. در روش CARLA نیز رسیدن به بهینه سراسری تضمین نمی‌گردد. در تحقیقات جامعی که در موسسه CalTech انجام شده است [8,9]، الگوریتمهای ژنتیکی (GA) با روشهای پیشرفته مبتنی بر سرد کردن فلزات (SA) مقایسه شده است و در اکثر موارد الگوریتمهای مبتنی بر SA نسبت به GA جوابهای بهتری را در مدت زمان کمتری پیدا کرده‌اند و در بقیه موارد کارایی این دو یکسان بوده است (SA همیشه سریعتر از GA بوده است). این آزمایشات بازای توابع محدب یا مقعر، توابع پیوسته یا گسسته، توابع خطی یا غیرخطی، توابع با تعداد پارامترهای زیاد یا کم، توابع با تعداد بهینه‌های محلی کم یا بسیار زیاد و حتی توابع

قطعی یا تصادفی (توابع هزینه همراه با نویز) انجام شده است. مواردی که جواب GA و SA یکسان بوده است، به ازای توابع هزینه‌ای بوده که دارای پارامترهای کم (۲ تا ۳ پارامتر) بوده و توابع مذکور پیوسته و دارای تعداد مینیمم‌های محلی کم بوده‌اند. نکته بسیار مهمی که در این آزمایشها مشاهده شد، این است که تفاوت کارایی GA در دفعات مختلف اجراء (Performance Variance) بیشتر از SA بوده است. البته در آزمایشهای مذکور از نسخه‌های پیشرفته و سریع SA استفاده شده است.

از طرف دیگر روشی مثل GA مخصوصاً برای مسائل با پارامترهای پیوسته دارای دقت و رزولوشن پایینی است. این پدیده می‌تواند به این دلیل باشد که در GA مقادیر حقیقی با رشته بیت‌ها کد می‌گردند و احتمال جهش برای بیت‌های کم‌ارزش و بیت‌های پرارزش یکسان است. یعنی در واقع احتمال تغییر بسیار زیاد یا تغییر بسیار کم مقادیر پارامترها یکسان است. در حالیکه در روشهایی مثل SA چنین نیست و با رسیدن به جوابهای بهتر و کاهش دما، رزولوشن نیز افزایش می‌یابد و احتمال تغییرات بسیار کم بیش از تغییرات زیاد است. در استراتژی‌های تکاملی (ES) و در برنامه‌ریزی تکاملی (EP) با بهره‌گیری از توزیعهای نرمال پیوسته، این ایراد برطرف شده است، ولی ایراد دیگری که وجود دارد، این است که در ES و EP مقدار رزولوشن یا واریانس توزیع یا همان اندازه گام (Step Size) بطور اتوماتیک تطبیق می‌یابد و سازگاری پیدا می‌کند و این سازگاری ممکن است بسیار کند باشد.

یکی از ایراداتی که به SA وارد است این است که از قابلیت توازی برخوردار نیست و به همین خاطر نسخه‌های موازی از SA پیشنهاد گردیده که این ایراد را برطرف می‌کند [10]. همچنین SA یک روش حافظه‌دار نمی‌باشد که باز هم روشهای حافظه‌دار مبتنی بر SA ابداع گردیده است [11]. هر چند همانطور که گفته شد نسخه‌های غیر موازی و بدون حافظه از SA بنا بر مشاهدات دارای کارایی بهتری نسبت به GA بوده‌اند [8,9].

با تکیه بر موارد ذکر شده در بالا در این تحقیق تصمیم گرفته شد بیشتر بر روی روشهای مبتنی بر SA تمرکز شود و این روشها بهینه گردند. یک روش جدید حافظه‌دار مبتنی بر SA نیز پیشنهاد گردیده است.

۳-۱- روش سرد کردن فلزات (SA)

یکی از روشهای جستجو و بهینه‌سازی سراسری که بر اصول ترمودینامیک مبتنی است و از دید آماری تضمین می‌کند که در صورت رعایت کردن روند کاهش دما (Temperature Schedule) بهینه سراسری را پیدا خواهد کرد، روش آبدادن فلزات یا روش سرد کردن فلزات (SA) است [12]. این روش کاربرد بسیار زیادی هم در مسائل گسسته و هم در مسائل پیوسته دارد [8,9]. طرز کار SA به این صورت است که در هر تکرار با استفاده از یک تابع چگالی احتمال تولید $g(x)$ ، یک نقطه تصادفی را در حول و حوش نقطه جواب قبلی، تولید می‌کند، تابع هزینه $E(x)$ را بازای این نقطه محاسبه می‌نماید و با استفاده از یک تابع احتمال پذیرش $h(x)$ نقطه جدید را بطور احتمالی قبول یا رد می‌نماید. اگر نقطه جدید دارای راندمان بهتری نسبت به نقطه قبلی باشد، احتمال قبول آن بیشتر است. البته در ابتدای جستجو که مقدار دما زیاد می‌باشد و محدوده جستجو وسیع است، احتمال قبول و رد نقطه جدید یکسان است (1/2). با کاهش دما و نزدیک شدن به نقاط بهینه پیدا شده، محدوده جستجو برای بالا بردن دقت و رزولوشن، کمتر می‌گردد. در بخشهای بعدی، روش SA و چند نسخه پیشرفته آن را ارائه خواهیم کرد.

۳-۱-۱- روش SA استاندارد یا BA

روش SA استاندارد یا BA به این صورت عمل می‌کند [8,9]:

۱- مقدار اولیه برای دما (T_1) و یک نقطه اولیه (x_1) انتخاب می‌شود.

$$x^* = x_1, \quad k=1 \rightarrow 2$$

۳- با استفاده از تابع چگالی احتمال تولید $g(x)$ ، یک نقطه جدید x بطور تصادفی در همسایگی x^* انتخاب می‌شود.

$$g(x, x^*) = (2\pi T_k)^{-\frac{D}{2}} \cdot \exp\left(-\frac{\|x - x^*\|^2}{2T_k}\right) \quad (12)$$

۴- تابع هزینه به ازای نقطه x محاسبه می‌شود $E(x)$.

۵- نقطه جدید x با احتمال $h(x)$ پذیرفته می‌شود (در صورت پذیرش $x^* = x$ است و در غیر اینصورت x^* تغییر نمی‌کند).

$$\Delta E = E(x) - E(x^*) \quad (13)$$

$$h(x) = \frac{1}{1 + \exp\left(\frac{\Delta E}{T_k}\right)} \quad (14)$$

$$k = k + 1 \quad ۶-$$

۷- دما کاهش می‌یابد.

$$T_k = \frac{T_1}{\ln k} \quad (15)$$

۸- اگر شرط خاتمه برقرار است، الگوریتم خاتمه می‌یابد و گرنه قدم ۳ اجراء می‌شود.

در الگوریتم فوق D تعداد پارامترهای تابع هزینه است و منظور از $\|x - x^*\|$ فاصله اقلیدسی دو بردار D بعدی x, x^* می‌باشد. تابع پذیرش یا $h(x)$ در بیشتر مواقع به شکل فوق است که به تابع Barker معروف است ولی به جای آن از تابع متروپولیس نیز می‌توان استفاده کرد [13]. در الگوریتم SA استاندارد، تابع چگالی احتمال تولید نقطه جدید از نوع گاوسی یا نرمال است و الگوریتم متناظر با چنین تابع چگالی احتمالی الگوریتم BA نامیده می‌شود. اگر روند کاهش دما (قدم ۷ از الگوریتم) از روند ذکر شده سریعتر نباشد، این روش تضمین می‌کند که از دید آماری حتماً بهینه سراسری را پیدا خواهد کرد. همانطور که دیده می‌شود، روش BA با روند کاهش دمای متناظر با آن (قدم ۷) بسیار کند است و روند رسیدن به جواب بسیار آهسته می‌باشد و به تکرارهای بسیار زیادی نیاز دارد. به همین دلیل نسخه‌های سریع و بسیار سریع از این الگوریتم پیشنهاد شده‌اند. اگر بخواهیم از روش SA برای بهینه‌سازی سراسری مدل پنهان مارکف استفاده کنیم، می‌توان تابع هزینه را چنین تعریف کرد:

$$E = -\text{Log } P(O|\lambda) \quad (16)$$

۳-۱-۲- روش SA سریع یا FA²⁰

همانطور که در قسمت قبلی ذکر شد، روند کاهش دما در الگوریتم SA استاندارد یا BA بسیار کند است و رسیدن به رزولوشن زیاد و دقت زیاد و دماهای کم برای حل مسائل پیوسته بسیار زمان بر خواهد بود. برای حل این مشکل یک الگوریتم سردکردن سریع یا FA پیشنهاد شده است [8,9] که برای تولید نقاط تصادفی جدید در حوض و حوش جواب قبلی از توزیع کوشی استفاده می‌کند. به همین خاطر این روش را روش CA²¹ نیز می‌نامند. یک تابع چگالی احتمال در فضای D بعدی از نوع کوشی برای تولید بردارهای تصادفی D بعدی بدین شکل می‌باشد [14]:

$$g(x, x^*) = \frac{\Gamma\left(\frac{D+1}{2}\right)}{\pi^{\frac{D+1}{2}}} \cdot \frac{T}{\left(\|x - x^*\|^2 + T^2\right)^{\frac{D+1}{2}}} \quad (17)$$

برای تضمین شرط رسیدن به نقطه بهینه سراسری لازم است که روند کاهش دما از روند زیر سریعتر نباشد:

$$T(k) = \frac{T_0}{k} \quad (18)$$

البته در روش FA روند کاهش دما از روند فوق نیز می‌تواند قدری سریعتر و بصورت زیر باشد [15]:

$$T(k) = \frac{T_0}{k^\alpha} \quad 1 \leq \alpha < 2 \quad (19)$$

همانطور که ملاحظه می‌گردد، روند کاهش دما در FA بسیار سریعتر از BA می‌باشد. در عمل چون الگوریتم سریعی برای تولید یک بردار تصادفی D بعدی با توزیع کوشی وجود ندارد و نیز به این دلیل که نیاز داریم دماهای بعدی مختلف از پارامترها متفاوت باشند، لذا در هر بعد از فضای D بعدی بطور مستقل از یک توزیع کوشی یک بعدی استفاده می‌شود که بصورت زیر است:

$$g_i(x_i, x_i^*) = \frac{1}{\pi} \cdot \frac{T_{ik}}{(x_i - x_i^*)^2 + T_{ik}^2} \quad (20)$$

برای تضمین شرط رسیدن به بهینه سراسری، باید روند زیر برای کاهش دما رعایت گردد.

$$T_i(k) = \frac{T_{i0}}{k^D} \quad (21)$$

۳-۱-۳ روش SA بسیار سریع یا VFA²²

در این روش [8,9] باز هم سعی خواهد شد که روند کاهش دما نسبت به روشهای قبلی تسریع گردد. اگر فرض کنیم x_{ik}^* (مقدار قبلی پارامتر بعد i ام در تکرار k ام) در فاصله $[A_i, B_i]$ قرار گیرد، مقدار جدید این پارامتر را در تکرار k+1 ام اینچنین تولید می‌کنیم:

$$x_{i(k+1)} = x_{ik}^* + y_i (B_i - A_i) \quad (22)$$

متغیر y_i یک متغیر تصادفی است که محدود و در فاصله $[-1, +1]$ می‌باشد و با تابع چگالی احتمال زیر تولید می‌شود:

$$g_i(y_i) = \frac{1}{2(|y_i| + T_{ik}) \ln\left(1 + \frac{1}{T_{ik}}\right)} \quad (23)$$

برای تضمین شرط رسیدن به بهینه سراسری، باید روند کاهش دما از روند زیر سریعتر نباشد:

$$T_i(k) = \frac{T_{i0}}{\exp\left(c_i k^{\frac{1}{D}}\right)} \quad (24)$$

پارامتر c_i که یک عدد مثبت است، برای کارکرد بهینه سیستم تنظیم می‌شود. همانطور که ملاحظه می‌گردد، روند کاهش دما در VFA بسیار سریعتر از BA, FA است و رسیدن به دقت و رزولوشن بالا در تعداد تکرار کمتری نسبت به روشهای قبلی میسر می‌گردد (البته با تضمین شرط رسیدن به بهینه سراسری).

۳-۲- یک روش SA جدید حافظه‌دار (MiGSA)

یکی از معایب روش SA و انواع مختلف آن که در بخش‌های قبلی معرفی گردید، این است که روش SA حافظه‌دار نمی‌باشد و این روش فقط یک نقطه از جوابهای قبلی را که بصورت احتمالی، بهترین نقطه می‌باشد، در حافظه خود ذخیره می‌کند. البته روشهایی مبتنی بر SA و از نوع حافظه‌دار پیشنهاد شده‌اند [11]. در اینجا یک روش SA حافظه‌دار و کاملاً جدید معرفی می‌گردد که در عین سادگی دارای کارایی بالاتری نسبت به روشهای قبلی است. در این روش جدید، ما از یک تابع چگالی احتمال مخلوط (Mixture PDF) برای تولید نقاط جدید استفاده می‌کنیم و به همین دلیل این روش را MiGSA (Mixture Generating function SA) نامیده‌ایم. این روش بصورت احتمالی M نقطه از بهترین جوابهای قبلی را در حافظه خود ذخیره می‌کند و تابع چگالی احتمال مخلوط $g(x)$ برای تولید نقطه جدید x به این صورت می‌باشد:

$$g(x) = \sum_{m=1}^M w_m g(x, x^{*(m)}) \quad (25)$$

در رابطه فوق $x^{*(m)}$ نقطه m ام از مجموعه M عضوی بهترین جوابهای قبلی می‌باشد و $g(x, x^{*(m)})$ یک تابع چگالی احتمال با بردار میانگین $x^{*(m)}$ است. این تابع چگالی احتمال می‌تواند هر کدام از توابع تولید بکار رفته در روشهای VFA, FA, BA باشد، ولی نکته قابل توجه این است که بازای هر کدام از انتخاب‌ها، باید روند متناظر آن را برای کاهش دما انتخاب کنیم. بعنوان مثال اگر تابع چگالی احتمال کوشی انتخاب شود، روند کاهش دما نیز باید روند متناظر با همان روش (FA) باشد. برای این که سطح زیر منحنی تابع چگالی احتمال مخلوط یک گردد، باید داشته باشیم:

$$\sum_{m=1}^M w_m = 1 \quad (26)$$

w_m را بصورت ابتکاری از رابطه زیر بدست می‌آوریم:

$$w_m = \frac{c_m}{\sum_{i=1}^M c_i} \quad (27)$$

$$c_m = 1 + \alpha \frac{E_{\max} - E(x^{*(m)})}{E_{\max} - E_{\min}} \quad (28)$$

E_{\max} مقدار هزینه به ازای بدترین عضو از مجموعه M عضوی S (مجموعه بهترین جوابهای قبلی) است. E_{\min} مقدار هزینه به ازای بهترین عضو از مجموعه S می‌باشد. همانطور که ملاحظه می‌شود، شانس بهترین عضو از این مجموعه $(1 + \alpha)$ برابر شانس بدترین عضو از این مجموعه می‌باشد. α یک عدد مثبت و بعنوان مثال برابر ۱ می‌تواند باشد. طرز عملکرد روش جدید MiGSA بصورت زیر است.

- ۱- دماهای اولیه برای بدهای مختلف T_{i0} انتخاب می‌شود.
- ۲- M نقطه اولیه مجموعه جوابهای S بصورت تصادفی انتخاب میشود.
- ۳- با استفاده از تابع چگالی احتمال مخلوط $g(x)$ ، یک نقطه جدید x را بطور تصادفی تولید می‌کنیم.
- ۴- تابع هزینه را بازای نقطه x محاسبه می‌کنیم $E(x)$.
- ۵- این نقطه را با احتمال پذیرش $h(x)$ می‌پذیریم. در صورت پذیرش، نقطه x جایگزین بدترین عضو از مجموعه M عضوی جوابهای قبلی (S) خواهد شد.

$$h(x) = \frac{1}{1 + \exp\left(\frac{\Delta E}{T_k}\right)} \quad (29)$$

$$\Delta E = E(x) - E_{\max} \quad (30)$$

۶- دما را با روند کاهش دمایی متناظر با تابع چگالی احتمال انتخاب شده، کاهش می‌دهیم.
 ۷- اگر شرط خاتمه برقرار است، الگوریتم پایان می‌یابد و گرنه به قدم ۳ خواهیم رفت.
 همانطور که ملاحظه می‌شود، در روش MiGSA هر چقدر تابع هزینه بازاری یکی از نقاط کمتر باشد، مقدار وزن آن و شانس آن بیشتر است و نیز اینکه اگر نقطه جدید پذیرفته شود، جایگزین بدترین نقطه از جوابهای قبلی خواهد شد. این خواص به اضافه خاصیت‌های بسیار خوب قبلی SA (از جمله تضمین رسیدن به بهینه سراسری)، روش MiGSA را برای بهینه‌سازی سراسری بسیار مناسب می‌سازد.

۳-۳- اتوماتون یادگیر تقویتی با مقدار عمل پیوسته (CARLA)

یکی از روشهایی که در بهینه‌سازی و کنترل بکار می‌رود، اتوماتون یادگیر یا LA^{23} می‌باشد [16]. اتوماتون یادگیر مدلی است که با یک محیط در ارتباط است و با توجه به ورودیهایی که از محیط یا سیستم دریافت می‌کند، عمل خود را با یک سیگنال تقویتی تصحیح نموده و عمل^{۲۴} جدید را به محیط یا سیستم اعمال می‌نماید. این کار مجدداً توسط محیط و سیستم ارزیابی می‌گردد و باز هم اتوماتون با دریافت ورودی از محیط یا سیستم عمل خود را تصحیح می‌کند. در مدلهایی که تا چندی قبل از اتوماتون‌های یادگیر پیشنهاد شده بودند، مقدار عمل اتوماتون یک مقدار گسسته بوده است. اخیراً یک روش اتوماتون یادگیر تقویتی با مقدار عمل پیوسته یا CARLA ارائه شده که می‌تواند برای بهینه‌سازی و کنترل در محیطهایی با پارامترهای پیوسته بکار رود [6,7]. CARLA در واقع یک مدل یادگیری تقویتی در محیطهای با پارامتر پیوسته می‌باشد. اساس روش CARLA اینگونه است که این روش مقدار عمل جدید را با استفاده از یک تابع چگالی احتمال عمل تولید می‌کند. اگر تعداد پارامترهای سیستم بیشتر از یک پارامتر باشد، تعدادی از اتوماتونهای یادگیر از نوع CARLA بطور موازی با سیستم در ارتباط خواهند بود و هر اتوماتون وظیفه بهینه‌سازی یک پارامتر از پارامترهای سیستم را به عهده خواهد داشت. این اتوماتونها با یکدیگر ارتباطی ندارند و فقط با محیط ارتباط دارند و در واقع ارتباط نسبی آنها از طریق محیط است. هر اتوماتون با استفاده از تابع چگالی احتمال عمل خود یک مقدار عمل جدید را تولید می‌کند، سیستم یا محیط مقدار عمل جدید را ارزیابی می‌کند و اتوماتون CARLA با استفاده از یک سیگنال تقویتی، تابع چگالی احتمال خود را تصحیح می‌نماید و این عمل تا رسیدن به نتیجه مطلوب ادامه می‌یابد. متأسفانه ابداع کنندگان این روش، اثباتی برای تضمین آماری پیدا کردن بهینه سراسری در کاربردهای بهینه‌سازی ارائه نداده‌اند. از جمله مزیت‌های این روش می‌توان به حافظه‌دار بودن این روش اشاره کرد که به نحوی اطلاعات نقاط قبلی جستجو را با در نظر گرفتن درجه اهمیت نسبی آنها، بعنوان حافظه سیستم نگهداری می‌کند. روش CARLA به صورت کلی چنین عمل می‌کند:

۱- ابتدائاً تابع چگالی احتمال عمل با یک تابع چگالی احتمال یکنواخت در محدوده پارامترهای سیستم مقدار دهی اولیه می‌شود.

۲- با توجه به تابع چگالی احتمال عمل، یک نقطه (عمل) جدید r را تولید می‌کند.

۳- پاسخ محیط (یا راندمان سیستم در کاربردهای بهینه‌سازی) را بدست می‌آورد.

۴- یک سیگنال تقویتی با توجه به مطلوبیت پاسخ سیستم محاسبه می‌شود.

۵- تابع چگالی احتمال عمل با توجه به این سیگنال تقویتی تصحیح می‌گردد.

۶- اگر شرط خاتمه برقرار نیست، قدم ۲ اجراء شده و گرنه الگوریتم خاتمه می‌یابد.

فرض کنید متغیر x در محدوده $[x_{\min}, x_{\max}]$ باشد. ابتدائاً تابع چگالی احتمال عمل بدین شکل و بصورت یک توزیع

یکنواخت می‌باشد:

$$f(x,1) = \frac{1}{x_{\max} - x_{\min}} \quad (31)$$

اگر در تکرار n ام باشیم، با استفاده از تابع چگالی احتمال عمل $f(x, n)$ ، عمل r را چنین انتخاب می‌کنیم:

$$F(r, n) = \int_{x_{\min}}^r f(x, n) dx = z(n) \quad (32)$$

یعنی ابتدائاً یک عدد تصادفی $z(n)$ با توزیع یکنواخت بین $[0,1]$ انتخاب می‌کنیم. آنگاه عمل r طوری انتخاب می‌شود که تابع احتمال تجمعی $F(r, n)$ برابر $z(n)$ گردد. تابع تجمعی $F(r, n)$ با استفاده از انتگرال گیری عددی در فاصله $[x_{\min}, r]$ انجام می‌شود. مقدار $f(x, n)$ را در نقاطی که نداریم، از طریق درونیایی خطی بدست می‌آوریم. با تولید عمل r به طریق فوق، این عمل را به محیط اعمال نموده و پاسخ محیط (هزینه) را $J(n)$ می‌نامیم. یک سیگنال تقویتی β برای تصحیح تابع چگالی احتمال عمل از رابطه زیر بدست می‌آید:

$$\beta(n) = \min \left(\max \left(0, \frac{J_{\text{med}} - J(n)}{J_{\text{med}} - J_{\text{min}}} \right), 1 \right) \quad (33)$$

سیگنال تقویتی β باید در فاصله $[0,1]$ باشد. $J_{\text{min}}, J_{\text{med}}$ به ترتیب میانه و میانگین هزینه بازای R نقطه از نقاط قبلی است (بعنوان مثال $R=500$). اگر تابع چگالی احتمال عمل در تکرار n ام $f(x, n)$ باشد، آنگاه تابع چگالی احتمال عمل برای تکرار بعدی ($n+1$ ام) به این صورت تصحیح می‌گردد:

$$f(x, n+1) = \begin{cases} \alpha [f(x, n) + \beta(n)H(x, r)] & \text{if } x \in [x_{\min}, x_{\max}] \\ 0 & \text{else} \end{cases} \quad (34)$$

که در آن $H(x, r)$ یک تابع گاوسی با میانگین r و به شکل زیر است:

$$H(x, r) = \frac{g_h}{(x_{\max} - x_{\min})} \cdot \exp \left(-\frac{1}{2} \cdot \frac{(x - r)^2}{(g_w (x_{\max} - x_{\min}))^2} \right) \quad (35)$$

که در آن پارامترهای g_w, g_h به ترتیب سرعت و رزولوشن (دقت) یادگیری را تعیین می‌کنند. می‌توان نوشت:

$$H(x, r) = \sqrt{2\pi} \cdot g_h g_w N(x, r, \sigma) \quad (36)$$

$$\sigma = g_w (x_{\max} - x_{\min}) \quad (37)$$

که $N(x, r, \sigma)$ یک تابع چگالی احتمال نرمال (گاوسی) با میانگین r و انحراف استاندارد σ میباشد. مقدار بهینه برای پارامترهای g_w, g_h در $[6,7]$ به ترتیب 0.3، 0.02 قرار داده شده‌اند. با این مقادیر، تصحیح تابع چگالی احتمال عمل بدین صورت خواهد بود:

$$f(x, n+1) = \begin{cases} \alpha [f(x, n) + 0.015 \beta(n) N(x, r, \sigma)] & \text{if } x \in [x_{\min}, x_{\max}] \\ 0 & \text{else} \end{cases} \quad (38)$$

در این روابط α ضریبی است که باید اعمال شود تا سطح زیر منحنی تابع چگالی احتمال عمل جدید یعنی $f(x, n+1)$ برابر یک شود. روابط فوق بدین معنی است که سطح زیر منحنی تابع چگالی احتمال گاوسی اضافه شده با فرض $g_w = 0.02, g_h = 0.3$ تقریباً باندازه 0.015 سطح زیر منحنی تابع چگالی احتمال $f(x, n)$ است (حداکثر و بازای $\beta = 1$). همانطور که قبلاً هم ذکر شد، در صورتیکه تعداد پارامترهای سیستم بیش از یک پارامتر باشد، تعدادی از اتوماتونهای CARLA بطور موازی با سیستم در ارتباط خواهند بود و هر CARLA وظیفه بهینه‌سازی یکی از پارامترهای سیستم را بر عهده دارد و در واقع ارتباط اتوماتونها با یکدیگر از طریق محیط است.

۴- روشهای ترکیبی برای بهینه‌سازی سراسری

یکی از مشکلات روشهای بهینه‌سازی سراسری این است که دقت و رزولوشن این روشها کم است. این گونه روشها در تکرارهای بسیار زیاد می‌توانند بهینه‌سازی را پیدا کنند، ولی در تکرارهای محدود و متوسط قادر به این کار نیستند. در واقع در روشهای جستجوی سراسری برداشتن قدم بعدی بصورت تصادفی است و بر اساس یک آینده‌نگری دقیق نمی‌باشد. در عوض روشهای جستجوی محلی قدمهای بعدی را براساس یک توجیه ریاضی و بصورت حساب شده بر می‌دارند. از این دید روشهای جستجوی محلی مثل BW بهتر از روشهای جستجوی محلی مثل حرکت در راستای عکس گرادیان (GD) است، زیرا در روش BW تضمین شده است که برداشتن گام بعدی باعث بهبود سیستم می‌گردد، ولی در روشی مثل GD این تضمین وجود ندارد، مگر اینکه نرخ یادگیری بطور دقیق تنظیم گردد که تنظیم دقیق آن برای تضمین بهبود سیستم کاری بسیار مشکل است. بطور خلاصه در روشهای جستجوی سراسری برداشتن گام بعدی بصورت تصادفی است و در یک راستای حساب شده در فضای بسیار-بعدي پارامترها نیست. از این رو روشهای جستجوی سراسری در تنظیم دقیق پارامترها (Fine Tuning) و داشتن دقت و رزولوشن بالا ضعیف هستند، در عوض می‌توانند از دام بهینه‌های محلی به راحتی بگریزند. از نقطه نظر دقت و رزولوشن نیز روشهای جستجوی سراسری متفاوت هستند. بعنوان مثال روشهای مبتنی بر SA همیشه با گذشت زمان دقت و رزولوشن خود را افزایش می‌دهند (واریانس توزیع کاهش می‌یابد)، ولی در روشهای مبتنی بر ES, EP مانند ES, EP دقت و رزولوشن سیستم بطور خودکار تنظیم شده و سازگاری پیدا می‌کند که این سازگاری ممکن است زمان زیادی بطول بیانجامد. روش GA هم دارای دقت کمتری نسبت به ES, EP برای تنظیم دقیق پارامترهاست [17]. در روشی مثل CARLA نیز دقت نسبی سیستم همواره یکسان است (برابر g_w است). روشهای جستجوی سراسری در تعداد تکرار بی‌نهایت بهینه‌سازی را پیدا خواهند کرد (در صورت تضمین آماری مانند روشهای مبتنی بر SA) ولی در تکرارهای محدود و متوسط قادر به این کار نیستند. از این رو محققین روشهای جستجوی ترکیبی را پیشنهاد کرده‌اند. در این روشها هم از قابلیت فرار از بهینه‌های محلی در روشهای سراسری، و هم از قابلیت تنظیم دقیق پارامترها و سرعت و دقت بالای جستجو در روشهای محلی استفاده می‌گردد. بعنوان مثال روشهای جستجوی سراسری مثل GA با روشهای محلی تلفیق شده‌اند [17, 18, 19, 20] و یا روشهای جستجوی سراسری مبتنی بر SA با روشهای جستجوی محلی توأم بکار رفته‌اند [21, 22]. روشهای ترکیبی برای جستجو و بهینه‌سازی را روشهای دو مرحله‌ای (Two-Phase) نیز می‌نامند [23]. دو مرحله‌ای بودن این روشها بدین معنی نیست که هر کدام از مراحل جستجوی محلی و جستجوی سراسری بطور مستقل و فقط یکبار انجام می‌گیرد، بلکه به دفعات مختلف و بصورت متناوب می‌توانیم در جستجوی ترکیبی بین انجام جستجوی محلی و جستجوی سراسری سوئیچ کنیم. برای انجام جستجوی ترکیبی می‌توانیم چند استراتژی را در پیش بگیریم:

۱- ابتدا با روشهای جستجوی سراسری یک نقطه اولیه خوب پیدا شود و با شروع از این نقطه و با استفاده از روشهای جستجوی محلی مقدار دقیق پارامترها را بدست آوریم.

۲- ابتدا یک جستجوی محلی انجام شود و پس از همگرا شدن جستجوی محلی، با استفاده از جستجوی سراسری این بهینه را بهینه‌تر کنیم.

۳- جستجوی محلی و جستجوی سراسری به کرات و با الگوریتمی که خواهیم گفت به صورت متوالی و متناوب انجام گیرد. استراتژی اول یک استراتژی منطقی نیست، زیرا همانطور که قبلاً گفتیم، در روشهای جستجوی محلی یک نقطه اولیه خوب ممکن است به نقطه نهایی بد همگرا گردد. این مساله را ما در آزمایشها برای روش جستجوی BW مشاهده کردیم و در مرجع [17] نیز برای روش محلی GD گزارش شده است. استراتژی دوم نیز مناسب نیست زیرا ممکن است ابتدائاً جستجوی محلی به یک نقطه بهینه محلی بد همگرا شود. نکته قابل توجه این است که هر کدام از استراتژیهای اول و دوم مانند روشهای جستجوی سراسری کند می‌باشند. هدف استراتژی سوم پیدا کردن یک نقطه شروع اولیه خوب است و به این ترتیب عمل می‌کند که مطابق با یک روند جستجوی سراسری (مثل SA) یک نقطه جدید را تولید می‌نماید. پس از تولید این نقطه (نقطه $x(n)$)، این نقطه را به عنوان نقطه شروع فرض کرده و یک جستجوی محلی را با شروع از این نقطه انجام می‌دهد. البته برای اینکه سرعت جستجوی ترکیبی بالا باشد، جستجوی محلی تا رسیدن به همگرایی و بطور کامل انجام نخواهد شد، بلکه به اندازه N_{LS} تکرار انجام می‌شود و در واقع یک جستجوی محلی تقریبی است و پس از انجام N_{LS} تکرار از جستجوی محلی، به نقطه $x_f(n)$ خواهیم رسید. آنگاه هزینه در نقطه $x_f(n)$ محاسبه می‌گردد و نقطه $x(n)$ را با استفاده از هزینه در نقطه $x_f(n)$ ارزیابی می‌کنیم و نقطه $x(n)$ را قبول یا رد می‌نماییم. در واقع هدف جستجوی ترکیبی از این نوع، آن است که یک نقطه در صورتی قبول شود که جستجوی محلی تقریبی با شروع اولیه از آن نقطه به جواب خوبی برسد. پس از قبول یا رد نقطه $x(n)$ بصورت احتمالی، مثلاً در روش SA دما کاهش می‌یابد و سپس نقطه بعدی $x(n+1)$ با استفاده از تابع چگالی احتمال تولید جدید، تولید می‌گردد. الگوریتم جستجوی ترکیبی ذکر شده بصورت زیر می‌باشد:

۱- انتخاب نقطه اولیه x_0 ($n=0$) و دمای اولیه T_0 .

۲- $n=n+1$

۳- تولید نقطه $x(n)$ با استفاده از تابع چگالی احتمال تولید.

۴- انجام جستجوی محلی (روش BW) باندازه N_{LS} تکرار با شروع از نقطه $x(n)$ و رسیدن به نقطه $x_f(n)$ پس از N_{LS} تکرار.

۵- بدست آوردن هزینه در نقطه $x_f(n)$.

۶- ارزیابی نقطه $x(n)$ با استفاده از مقدار هزینه در نقطه $x_f(n)$ و قبول یا رد نقطه $x(n)$.

۷- کاهش دما.

۸- اگر شرط خاتمه برقرار نیست، به قدم ۲ می‌رویم و گرنه الگوریتم پایان می‌یابد.

همانطور که در بخش آزمایشات نیز خواهیم دید، در اکثر مواقع در روشهای جستجوی محلی مانند BW خوب بودن یا نبودن یک نقطه اولیه در همان تکرارهای اولیه معلوم می‌گردد و برای ارزیابی یک نقطه اولیه لزومی ندارد که جستجوی محلی را تا پایان و تا رسیدن به همگرایی انجام دهیم. در واقع تلفیق روشهای جستجوی محلی و جستجوی سراسری چندین مزیت را به همراه دارد:

۱- سریع شدن جستجو و بهینه‌سازی نسبت به جستجوی سراسری به تنهایی و رسیدن به نقطه بهتر در تعداد تکرارهای کمتر.

۲- افزایش دقت و رزولوشن تخمین پارامترها با استفاده از قدرت جستجوی محلی.

۳- جلوگیری از جستجوی کور کورانه.

۴- به دست آوردن راندمان بالا در جستجو (بعنوان مثال افزایش راندمانی که با چند تکرار از جستجوی محلی BW بدست می‌آید (۱۰ تکرار) ممکن است در هزاران تکرار از جستجوی سراسری به تنهایی بدست نیاید، زیرا حرکت در روشهای جستجوی محلی حساب شده است).

نکته‌ای که باید به آن توجه داشت این است که تعداد تکرارهای جستجوی محلی نباید از یک حدی کمتر باشد، زیرا در اینصورت تخمین ما و ارزیابی ما در مورد نقطه شروع اولیه $x(n)$ در هر تکرار درست نخواهد بود.

۵- آزمایشها

در این بخش به ارائه نتایج بدست آمده در آزمایشهای انجام شده خواهیم پرداخت. هدف از این آزمایشها آموزش مدل پنهان مارکف با معیار بیشترین شباهت (ML) و توسط روشهای محلی، سراسری و ترکیبی می باشد. مدل مارکفی که قرار است آموزش داده شود، یک مدل پنهان مارکف پیوسته با سه حالت و سه تابع گاوسی در هر حالت با ماتریسهای کوواریانس قطری می باشد. اگر تعداد پارامترهای سیستم با فرض قطری بودن ماتریس کوواریانس P باشد، داریم:

$$P=N(1+N+M(2D+1)) \quad (39)$$

که به ازای $D=2, M=3, N=3$ تعداد پارامترهای مدل برابر ۵۷ پارامتر خواهد شد. آموزش این مدل در حالت رشته‌های مشاهدات چندتایی (Multiple Observation) بوده و برای آموزش آن از ۱۰ رشته مشاهده با طول $T=20$ استفاده شده است. هدف، بیشینه کردن $P(O|\lambda)$ است که O مجموعه ۱۰ رشته مشاهده و λ پارامترهای مدل پنهان مارکف می باشد. همانطور که می دانیم آموزش مدل پنهان مارکف یک مسأله بهینه‌سازی مقید است، زیرا باید شروط زیر برقرار باشد:

$$v_{ii} > 0 \quad \text{و} \quad \sum_{m=1}^N c_{im} = 1 \quad \text{و} \quad \sum_{j=1}^N a_{ij} = 1 \quad (40)$$

v_{ii} برابر با عنصر قطری ماتریس کوواریانس و متناظر با واریانس در بعد i ام است. برای تبدیل مسأله مقید HMM به یک مسأله نامقید برای بهینه‌سازی از نگاشتهای ابتکاری زیر استفاده کرده‌ایم:

$$a_{ij} = \frac{\exp(a'_{ij})}{\sum_{j=1}^N \exp(a'_{ij})} \quad (41)$$

$$c_{im} = \frac{\exp(c'_{im})}{\sum_{m=1}^M \exp(c'_{im})} \quad (42)$$

$$v_{ii} = \exp(v'_{ii}) \quad (43)$$

این روابط بدان معنی است که روشهای بهینه‌سازی سراسری نامقید همچون SA، پارامترهای سیستم را در فضای $\pi'_i, a'_{ij}, c'_{im}, v'_{ii}$ بصورت نامقید تغییر می دهند، ولی تبدیل شده این پارامترها توسط روابط بالا قیود و شروط مدل مارکف را اقلع میکنند. حال به شرح آزمایشها خواهیم پرداخت.

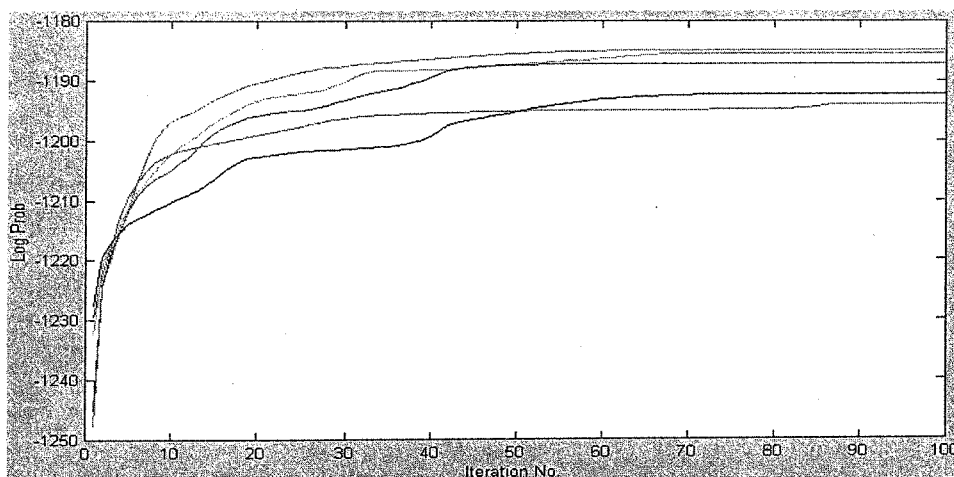
۵-۱- روش آموزش BW

همانطور که قبلاً ذکر شد روش BW یک روش جستجوی محلی است و پر کاربردترین روش آموزش محلی مدلهای HMM می باشد. در این قسمت به ازای داده‌های آموزشی مذکور در قسمت قبل، در شکل (۲) از چهار نقطه تصادفی مختلف شروع کرده‌ایم و تا رسیدن به همگرایی تکرارها را ادامه داده‌ایم. مقدار لگاریتم احتمال در نقطه شروع و در نقطه همگرایی در هر کدام از چهار حالت در جدول (۱) ذکر گردیده است. نقطه قابل توجه و مهمی که وجود دارد این است که شروع از یک نقطه اولیه بد ممکن است به یک نقطه همگرایی (نقطه نهایی) خوب منجر شود. این پدیده در مورد روش آموزش بر اساس حرکت در

راستای عکس گرادیان (GD) هم در مرجع [17] گزارش شده است. در واقع همین پدیده دید ما را در مورد طراحی الگوریتم‌های جستجو مخصوصاً الگوریتم‌های جستجوی ترکیبی به شدت تحت تأثیر قرار می‌دهد.

جدول (۱) لگاریتم احتمال بازای نقطه شروع و نقطه همگرایی در چهار آزمایش مختلف از روش BW.

آزمایش اول	آزمایش دوم	آزمایش سوم	آزمایش چهارم	
-1229.3	-1232.4	-1245.2	-1227.4	نقطه شروع
-1193.4	-1185.4	-1187.0	-1192.2	نقطه همگرایی

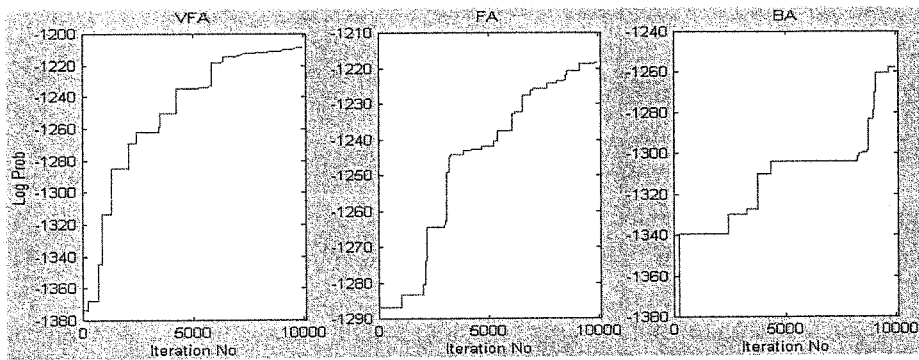


شکل (۲) نمودار لگاریتم احتمال بر حسب تعداد تکرارها در چهار آزمایش مختلف از انجام روش BW.

همانطور که مشاهده می‌شود با شروع از نقاط شروع مختلف به نقاط همگرایی متفاوتی رسیده‌ایم (داده‌های آموزشی در چهار آزمایش یکسان است) و در واقع به بهینه‌های محلی مختلف همگرا شده‌ایم که این ایراد اصلی روش‌های آموزش (جستجوی) محلی است که با استفاده از روش‌های جستجوی سراسری و ترکیبی درصد رفع این ایراد هستیم. نقطه قابل توجه دیگر این است که سرعت روش BW در چند تکرار اول بسیار زیاد است و منحنی لگاریتم احتمال دارای شیب تندی در ابتدای آموزش می‌باشد.

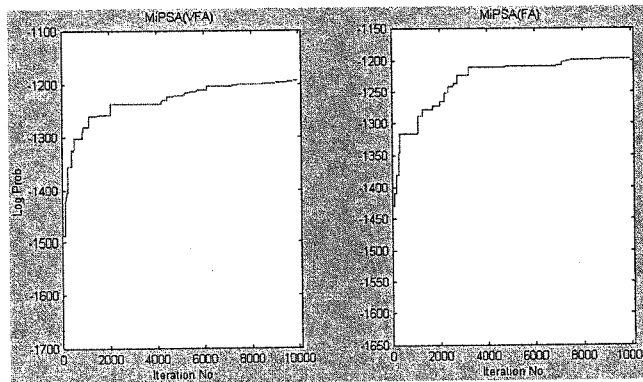
۵-۲- روش‌های مبتنی بر سرد کردن فلزات (SA)

از روش‌های مبتنی بر SA در این قسمت از سه روش VFA, FA, BA استفاده کردیم. در این آزمایش‌ها، قبول یا رد نقطه جدید بطور قطعی بوده است (Deterministic) نه بصورت احتمالی. یعنی اگر هزینه در نقطه جدید کمتر بوده است، نقطه پذیرفته شده و در غیر اینصورت رد می‌شود. مقادیر Max, Min بعضی پارامترها (در جاهایی مثل VFA که نیاز داریم) و همچنین دماهای اولیه در ابعاد مختلف را با استفاده از اطلاعاتی چون کمترین و بیشترین مقدار پارامترها در بعدهای مختلف، میانگین کل داده‌ها و واریانس کل داده‌ها در بعدهای مختلف بدست آورده‌ایم و مقدار دهی اولیه کرده‌ایم. شکل (۳) نمونه‌ای از عملکرد هر یک از سه روش VFA, FA, BA را نشان می‌دهد. همانطور که دیده می‌شود تغییرات لگاریتم احتمال در انتهای جستجو در VFA, FA کم است و این به آن دلیل است که در این روش‌ها در انتهای جستجو محدوده جستجو بسیار کوچک شده (رزولوشن و دقت جستجو زیاد شده) و تغییرات تابع هزینه کم می‌شود، ولی در روشی مثل BA بدلیل اینکه روند کاهش دما بسیار کند است، حتی در ۱۰۰۰۰ تکرار هم واریانس توزیع و دمای آن هنوز هم زیاد بوده و محدوده تغییرات پارامترهای سیستم زیاد است و تغییرات تابع هزینه هنوز هم زیاد می‌باشد.



شکل (۳) نمونه‌ای از عملکرد سه روش VFA, FA, BA.

در شکل (۴) نیز نمونه‌ای از عملکرد روش SA حافظه‌دار (MiPSA) نشان داده شده است. در یکی از اشکال نمونه‌ای از عملکرد روش FA حافظه‌دار و در شکل دیگر عملکرد روش VFA حافظه‌دار ترسیم گردیده است. حافظه سیستم در این دو حالت یعنی M برابر ۱۰ اختیار شده است. یعنی سیستم ۱۰ نقطه از بهترین نقاط بدست آمده قبلی را در حافظه خود نگه می‌دارد.

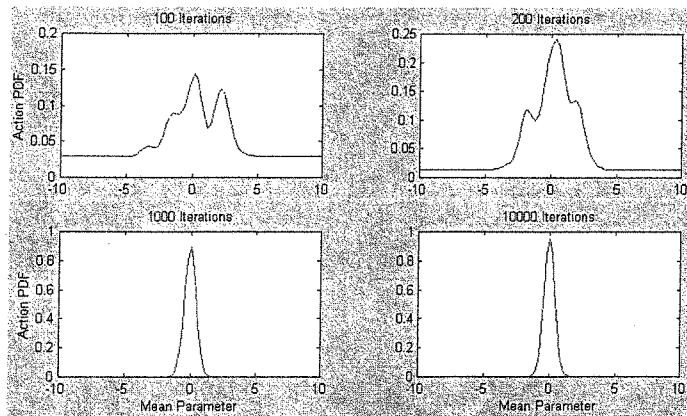


شکل (۴) نمونه‌ای از عملکرد روش FA حافظه‌دار و VFA حافظه‌دار.

۵-۳- روش CARLA

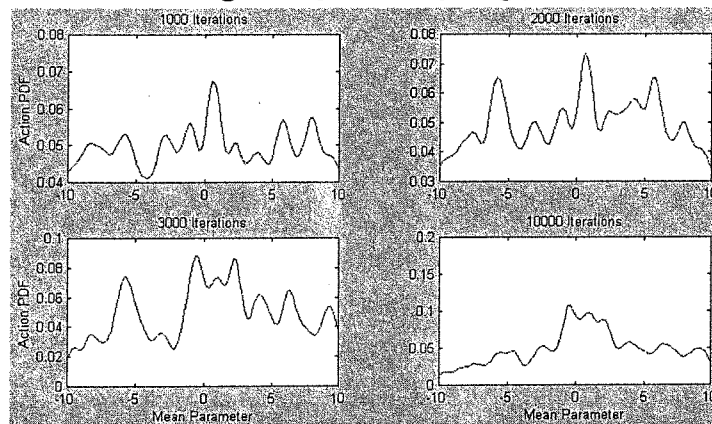
همانطور که قبلاً ذکر شد در روش CARLA، هر اتوماتون یکی از پارامترهای مدل پنهان مارکف را بهینه می‌کند و جمعیتی از اتوماتونهای CARLA (به تعداد پارامترهای مدل مارکف) بصورت موازی و مستقل از همدیگر پارامترهای سیستم را بهینه می‌کنند. مقادیر می‌نیمم و ماکزیمم پارامترهای مدل پنهان مارکف از روی داده‌های آموزشی بدست می‌آیند. هر یک از اتوماتونهای CARLA در هر تکرار از آموزش (جستجو) سعی می‌کند تابع چگالی احتمال عمل خود را تصحیح کند و در نهایت شکل تابع چگالی احتمال تقریباً گاوسی است که مرکز آن بر روی نقطه بهینه سراسری قرار دارد. اگر دقت نسبی CARLA برابر ϵ باشد، تابع چگالی احتمال نهایی در صورت همگرا شدن روش، بر روی نقطه بهینه قرار خواهد گرفت، ولی انحراف استاندارد این تابع گاوسی بیش از ϵ خواهد بود (همانطور که در آزمایشات [7] دیده شده است).

در آزمایش اول ابتدا CARLA را برای آموزش یک مدل پنهان مارکف با تعداد پارامترهای کم بکار می‌بریم. در این آزمایش مسأله بهینه‌سازی و آموزش یک مدل پنهان مارکف با یک حالت و یک تابع گاوسی در آن حالت و در فضای یک بعدی میباشد. پارامترهای مدل مقدار میانگین و واریانس تابع گاوسی است و تعداد پارامترهای مدل برابر ۲ می‌باشد. شکل (۵) شکل تابع چگالی احتمال عمل را در تکرارهای مختلف نشان می‌دهد. پارامتر مورد نظر در این شکل‌ها میانگین تابع گاوسی بوده است که مقدار واقعی آن صفر بوده و CARLA به همان مقدار واقعی همگرا شده است و تابع چگالی احتمال در نهایت به یک تابع گاوسی تبدیل شده است.



شکل (۵) شکل تابع چگالی احتمال عمل در بعد اول از یک مدل مارکف با دو پارامتر.

در آزمایش دوم از CARLA برای بهینه‌سازی یک پنهان مارکف با $D=2$, $M=3$, $N=3$ استفاده شده است. داده‌های آموزشی همان داده‌های آموزشی استفاده شده در روشهای SA و BW است. تعداد پارامترهای سیستم 57 پارامتر می‌باشد که نسبتاً زیاد است. شکل (۶) شکل تابع چگالی احتمال عمل را در تکرارهای مختلف نشان می‌دهد. همانگونه که دیده می‌شود، تابع چگالی احتمال عمل حتی در تکرارهای زیاد هم به خوبی همگرا نشده است و آشفته می‌باشد.



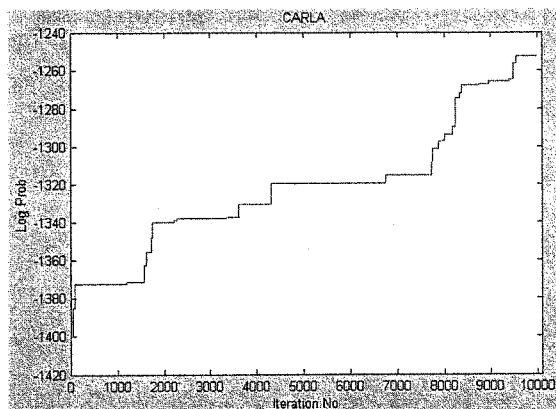
شکل (۶) شکل تابع چگالی احتمال عمل برای پارامتر بعد اول از یک مدل مارکف با ۵۷ پارامتر.

CARLA برای بهینه‌سازی مسائل با تعداد پارامترهای ۲ و ۳ و با تعداد می‌نیم‌های محلی کم (۲ می‌نیم محلی) کارایی خوبی داشته است [6,7] و ما نیز در آزمایش قبل آن را مشاهده کردیم، ولی کارایی آن برای مسائل با تعداد پارامترهای متوسط یا زیاد پایین بوده و روند بهبود آن کند است. این پدیده می‌تواند به چند دلیل باشد:

۱- با فرض اینکه β عددی بین صفر و یک باشد [6,7]، یک نقطه جدید بسیار خوب (نسبت به نقطه‌های قبلی) نمی‌تواند تأثیر زیادی در شکل تابع چگالی احتمال عمل داشته باشد و با فرض $g_h = 0.3$ و $g_w = 0.02$ میزان تأثیرگذاری این نقطه جدید بسیار خوب در شکل تابع چگالی احتمال عمل بطور نسبی حداکثر برابر با 0.015 خواهد بود (مطابق با فرمولهای ذکر شده). البته اگر هم حداکثر مقدار β به یک محدود نگردد، پایداری تابع چگالی احتمال عمل کمتر خواهد شد.

۲- در صورت همگرایی و با مقدار دقت g_w ، تابع چگالی احتمال نهایی دارای انحراف استاندارد بیشتری از g_w است [7]. اگر مقدار g_w را افزایش دهیم، دقت سیستم پایین خواهد آمد و اگر مقدار g_w را کاهش دهیم، دقت سیستم افزایش خواهد یافت ولی شکل تابع چگالی احتمال دیرتر و کندتر همگرا و پایدار خواهد شد.

شکل (۷) نمونه‌ای از عملکرد CARLA را در آموزش مدل مارکف با ۵۷ پارامتر نشان می‌دهد.



شکل (۷) نمونه‌ای از عملکرد CARLA در آموزش مدل مارکف با ۵۷ پارامتر.

۵-۴. مقایسه‌ای بین روشهای محلی و سراسری

در این بخش هدف ما مقایسه روشهای محلی و سراسری در بهینه‌سازی و آموزش مدل پنهان مارکف است. در هر کدام از آزمایشها، تعداد تکرارها ۱۰۰۰۰ تکرار بوده است (غیر از BW). برای آنکه نتایج مقایسه یک نتیجه قابل تعمیم و سازگار (Consistent) باشد، هر کدام از روشها به دفعات انجام شده‌اند و شرایط آزمایش برای روشهای مختلف یکسان بوده است. به این منظور برای هر روش، آموزش سیستم ۵۰ بار انجام شد. یعنی ۵۰ نقطه تصادفی برای شروع اولیه و ۵۰ مجموعه آموزشی مختلف، طوری که ۵۰ نقطه شروع اولیه و نیز ۵۰ مجموعه آموزشی مختلف برای تمام روشها یکسان باشد. پس از این مرحله نیز در تولید اعداد تصادفی، بذر اولیه (Seed) برای تمام روشها یکسان قرار داده شد تا شرایط آزمایش کاملاً مساوی باشد. هر آزمایش تا تعداد تکرار برابر با ۱۰۰۰۰ تکرار انجام شد. پس از اتمام آزمایشها، میانگین لگاریتم احتمال بر روی ۵۰ آزمایش را بعنوان میانگین کارایی روش معیار قرار دادیم. جدول (۲) متوسط راندمان کارایی (متوسط لگاریتم احتمال) را به ازای روشهای مختلف نشان می‌دهد.

جدول (۲) مقایسه راندمان متوسط روشهای جستجوی محلی و سراسری در ۵۰ آزمایش مختلف.

BW	MiPSA(VFA)	MiPSA(FA)	VFA	FA	CARLA	BA
-1189.6	-1195.4	-1198.5	-1210.3	-1215.7	-1250.4	-1259.1

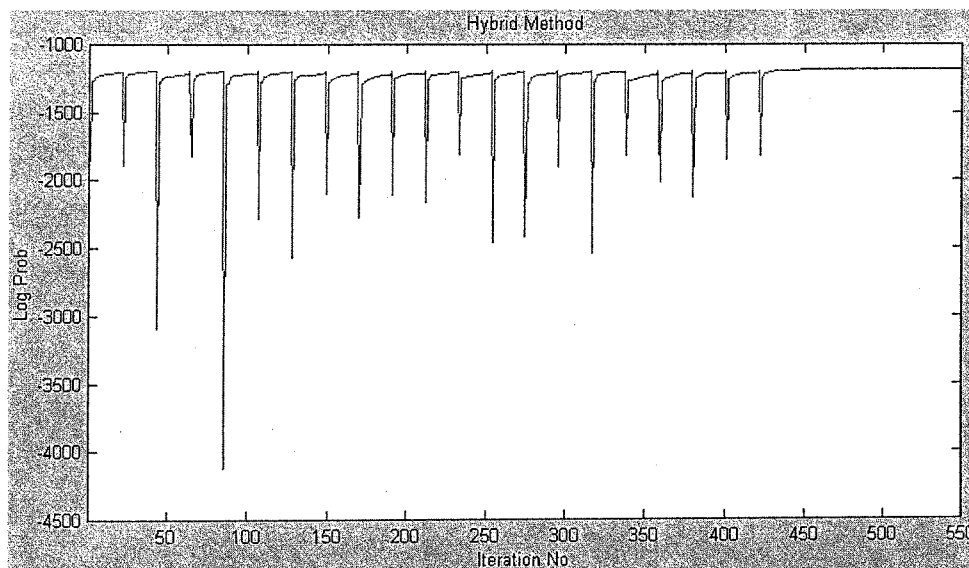
همانطور که ملاحظه می‌گردد، روش محلی BW هنوز هم بیشترین راندمان را داراست. این می‌تواند بدان دلیل باشد که سرعت و دقت روش BW بسیار بالاست و نیز در هر تکرار از الگوریتم BW این تضمین وجود دارد که کارایی سیستم بهبود خواهد یافت. البته اگر تعداد تکرارها بیش از ۱۰۰۰۰ گردد، روشهای سراسری از یک جایی به بعد مطمئناً از روش BW سبقت خواهند گرفت، ولی فعلاً هدف ما آموزش سیستم در تعداد تکرارهای نه خیلی زیاد می‌باشد. دیده می‌شود که راندمان هر یک از روشهای SA حافظه‌دار (MiPSA) بیش از روشهای SA معمولی نظیر آن می‌باشد و این به دلیل حافظه‌دار بودن روشهای MiPSA می‌باشد. همانطور که ملاحظه می‌شود، راندمان روشهای FA, VFA بیش از روش CARLA است و این بدان دلیل است که اولاً رسیدن به نقطه بهینه در این روشها بطور آماری تضمین شده است و ثانیاً دقت و رزولوشن در این روشها بسیار بیشتر از روشی مثل CARLA است و روش CARLA نسبت به FA, VFA کندتر است. راندمان BA از راندمان CARLA کمتر است. این می‌تواند به دلیل حافظه‌دار بودن روش CARLA و نیز کم بودن دقت روش BA در ۱۰۰۰۰ تکرار باشد، زیرا روند کاهش دما در BA بسیار کند است و در ۱۰۰۰۰ تکرار نمی‌تواند به دقت و رزولوشن بالایی برسد (به دماهای پایین برسد). به همین دلیل راندمان روش BA پایین‌تر از روشهای FA, VFA است. در روشهای FA, VFA تابع چگالی احتمال تولید در نقطه فعلی تیز است و دارای دقت بالایی است. در VFA شکل تابع چگالی احتمال تولید در نقطه فعلی تیزتر از شکل تابع در روش FA است و دقت آن بیشتر است. اصطلاحاً این نوع از توابع چگالی احتمال برخلاف تابع چگالی احتمال نرمال دارای یک قلّه تیز و در عین حال حواشی سنگین یا دم سنگین هستند (Heavy-Tailed). قلّه تیز باعث بالا رفتن دقت جستجو می‌شود و حواشی سنگین باعث فرار راحت‌تر از بهینه‌های محلی می‌گردد.

۵-۵ - روش ترکیبی

در این بخش، از الگوریتم جستجوی ترکیبی ذکر شده استفاده خواهیم کرد. روش جستجوی سراسری استفاده شده در این قسمت، روش VFA حافظه‌دار یا MiPSA(VFA) است. تعداد تکرارهای جستجوی سراسری ۱۰۰۰ تکرار انتخاب شد و تعداد تکرارهای جستجوی محلی BW در هر بار تکرار از جستجوی سراسری 5, 10, 15 تکرار اختیار شد. نتایج آزمایشها در جدول (۳) نشان داده شده است. شکل (۸) نمونه‌ای از عملکرد روش ترکیبی را نشان می‌دهد.

جدول (۳) کارایی متوسط روش ترکیبی بازای تعداد تکرارهای مختلف در جستجوی محلی.

N_{LS} (تعداد تکرارهای BW)	5	10	15
کارایی متوسط	-1203.4	-1181.3	-1180.9



شکل (۸) نمونه‌ای از عملکرد روش ترکیبی برای آموزش مدل مارکف.

همانطور که ملاحظه می‌شود اگر N_{LS} بیش از ۱۰ تکرار باشد، راندمان جستجوی ترکیبی بیش از راندمان روش BW است. می‌بینیم که اگر N_{LS} ، ۵ تکرار باشد، راندمان کمتر از BW است. این بدان دلیل است که با ۵ تکرار از جستجوی محلی BW، نمی‌توان یک ارزیابی تقریبی درست در مورد نقطه فعلی داشت و تعداد تکرارهای جستجوی محلی BW حداقل باید ۱۰ تکرار باشد. اگر در روش جستجوی ترکیبی ۱۰۰۰ تکرار از جستجوی سراسری انجام گیرد و در هر تکرار ۱۰ تکرار از جستجوی محلی انجام شود، آنگاه تعداد تکرارهای کل، برابر همان ۱۰۰۰۰ تکرار در آزمایشهای قبلی مربوط به روشهای سراسری خواهد شد (البته یک تکرار از جستجوی محلی BW بیش از یک تکرار از جستجوی سراسری زمان می‌گیرد و بیشتر طول می‌کشد). در شکل (۸) نمونه‌ای از عملکرد روش ترکیبی مشاهده می‌گردد. در واقع در این روش ترکیبی با کمک روشهای مبتنی بر SA و استفاده از روش محلی BW در یک تعداد تکرار نه خیلی زیاد به راندمانی بیشتر از راندمان روشهای محلی BW رسیده‌ایم. راندمان متوسط در حالت $N_{LS} = 15, N_{LS} = 10$ نزدیک به هم است و شاید این بیانگر آن باشد که ما به بهینه سراسری واقعی نزدیک شده‌ایم.

قابل توجه آنکه روشهایی مثل SA بسیار سریعتر از روشهایی مثل GA، هستند [8,9]؛ زیرا در روشهای مبتنی بر SA در هر تکرار یک نقطه (یک راه حل) تولید و ارزیابی می‌گردد، در حالیکه در روشهایی مثل GA، ES یا EP در هر تکرار، جمعیتی از نقاط (راه‌حل‌ها) باید تولید و ارزیابی شوند.

۶- نتیجه گیری

در این مقاله برای آموزش مدل پنهان مارکف، از روشهای جستجوی محلی، جستجوی سراسری و جستجوی ترکیبی استفاده شد. از میان روشهای جستجوی سراسری، روشهای مبتنی بر SA و روش CARLA مبتنی بر اتوماتون یادگیر با عمل پیوسته استفاده شدند. آزمایشها نشان دادند که روشهای پیشرفته مبتنی بر SA بهتر از روش CARLA جواب میدهد ولی نسخه استاندارد SA یعنی BA دارای کارایی پائین تری نسبت به CARLA است. همچنین یک روش کاملاً جدید و حافظه دار مبتنی بر SA بنام MiGSA پیشنهاد و استفاده گردید که نسبت به روشهای SA معمولی و پیشرفته (بدون حافظه) دارای کارایی بالاتری است. کارایی متوسط روش محلی BW از روشهای سراسری فوق در آزمایشهای انجام شده در تعداد تکرارهای نه خیلی زیاد (۱۰۰۰۰ تکرار) بالاتر بوده است و این بدلیل سرعت بالای روش BW، دقت فوق العاده آن و مبنای ریاضی مستحکم این روش است. با استفاده از ترکیب روشهای جستجوی محلی (BW) و روشهای جستجوی سراسری (مبتنی بر SA)، در تعداد تکرارهای نه خیلی زیاد، به کارایی متوسط بیشتری نسبت به روش محلی BW رسیدیم. روش ترکیبی استفاده شده هم از خاصیت جستجوی وسیع در روشهای سراسری استفاده می کند (فرار از بهینه های محلی) و هم از سرعت، دقت و رزولوشن بالای جستجوی محلی بهره می گیرد.

زیر نویس ها

- | | |
|---|------------------------------|
| 1- Hidden Markov Model | 14- Global Optimum |
| 2- Baum-Welch | 15- Global Search |
| 3- Local | 16- Global Optimization |
| 4- Simulated Annealing | 17- Genetic Algorithms |
| 5- Continuous Action Reinforcement Learning | 18- Evolutionary Strategies |
| 6- Automata | 19- Evolutionary Programming |
| 7- Boltzmann Annealing | 20- Generating Function |
| 8- Mixture Generating function Simulated | 21- Acceptance Function |
| 9- Annealing | 22- Fast Annealing |
| 10- Hybrid | 23- Cauchy Annealing |
| 11- Maximum Likelihood | 24- Very Fast Annealing |
| 12- Expectation Maximization | 25- Learning Automata |
| 13- Gradient Descent | 26- Action |

مراجع

- [1] Rabiner, L.R., Juang, B.-H., Fundamentals of Speech Recognition, Prentice-Hall, 1993.
- [2] McLachlan, G., Peel, D., Finite Mixture Models, John Wiley & Sons, 2000.
- [3] Figueiredo, M.A.T., Jain, A.K., "Unsupervised Learning of Finite Mixture Models", IEEE Trans. on PAMI, Vol. 24, No. 3, pp. 381-896, March 2002.
- [4] Bäck, T., Evolutionary Algorithms in Theory and Practice, Oxford University Press, 1996.
- [5] Stuckman, B.E., Eason, E.E., "A Comparison of Bayesian/Sampling Global Optimization Techniques", IEEE Trans. on SMC, Vol. 22, No. 5, Sep. 1992.
- [6] Howell, M.N., Gordon, T.J., Best, M.C., "The Application of Continuous Action Reinforcement Learning Automata to Adaptive PID Tuning", --- .
- [7] Howell, M.N., Gordon, T.J., "Continuous Learning Automata and Adaptive Digital Filter Design", UKACC International Conference on Control, Sep. 1998.
- [8] Ingber, L., Rosen, B., "Genetic Algorithms and Very Fast Simulated Reannealing: A Comparison", Mathematical and Computer Modelling, Vol. 16, No. 11, 1992.
- [9] Ingber, L., "Simulated Annealing: Practice versus Theory", Mathematical and Computer Modelling, Vol. 18, No. 11, 1993.
- [10] Kimura, K., Taki, K., "Time-Homogeneous Parallel Annealing Algorithm", Technical Report, Institute for New Generation Computer Technology, Tokyo, Japan, 1992.
- [11] Lo C.-C., Hsu, C.-C., "An Annealing Framework with Learning Memory", IEEE Trans. on SMC, Vol. 28, No. 5, Sep. 1998.
- [12] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., "Optimization by Simulated Annealing", Science, Vol. 220, No. 4598, pp. 671-680, May 1983.



- [13] Locatelli, M., "Simulated Annealing Algorithms for Continuous Global Optimization", in Horst, R., Pardalos, P.M.(Eds.), Handbook of Global Optimization, Kluwer Academic Publishers, Boston, 1995.
- [14] Lindsey, J.K., Lindsey, P.J., "Multivariate Distributions with Correlation Matrices for Nonlinear Repeated Measurements", --- .
- [15] Fausett, L.V., Fundamentals of Neural Networks, Prentice-Hall, 1994.
- [16] Narendra, K.S., Thathatchar, M.A.L., Learning Automata: An Introduction, Prentice-Hall, 1989.
- [17] Yao, X., "Evolving Artificial Neural Networks", Proceedings of the IEEE, Vol. 87, No. 9, pp. 1423-1447, Sep. 1999.
- [18] Bäck, T., Fogel, D., Michalewics, Z.(Eds.), Handbook of Evolutionary Computation, Institute of Physics Publishing & Oxford University Press, 1997.
- [19] Hart, W.E., Adaptive Global Optimization with Local Search, PhD Thesis, University of California at San Diego, 1994.
- [20] Renders, J.-M., Flasse, S.P., "Hybrid Methods using Genetic Algorithms for Global Optimization", IEEE Trans. on SMC, Vol. 26, No. 2, Apr. 1996.
- [21] Desai, R., Patil, R., "SALO: Combining Simulated Annealing and Local Optimization for Efficient Global Optimization", Proceedings of the 9th Florida AI Research Symposium (FLAIRS-96), pp. 233-237, 1996.
- [22] Lee, Y.-O. et al., "Optical Model Parameter Search with Simulated Annealing and Marquardt-Levenberg Method", Symposium on Nuclear Data, Japan, 2000.
- [23] Schoen, F., "Two-Phase Methods for Global Optimization", in Horst, R., Pardalos, P.M.(Eds.), Handbook of Global Optimization, Kluwer Academic Publishers, Boston, 1995.