

# کاربرد فنون فرا ابتکاری در برنامه‌سازی خودکار

علیرضا خلیلیان، دانشجوی دکتری نرم‌افزار

گروه مهندسی نرم‌افزار، دانشکده مهندسی کامپیوتر، دانشگاه اصفهان

[khalilian@eng.ui.ac.ir](mailto:khalilian@eng.ui.ac.ir)

**چکیده:** سال‌هاست که دانشمندان تحقیق می‌کنند چطور برنامه‌سازی رایانه را به خود رایانه محول کنند! ما به رایانه بگوییم چه می‌خواهیم و او برنامه‌اش را بسازد و بعد اجرا کند. مقاله حاضر قصد دارد خواننده را فقط با مفاهیم و روش‌های کلی این مسئله تحقیقاتی آشنا کند و سرنخ‌هایی از شیوه‌های حل آن ارائه نماید.

## ۱-۱- مقدمه

از ابتدای پیدایش رایانه‌ها، برنامه‌سازی توسط انسان انجام شده است و این روند تا سال‌های آینده هم قطعاً ادامه دارد. اما چندین سال است که محققان به فکر افتادند که چگونه می‌توان برنامه‌سازی را خودکار کرد؟ یعنی هدف اینست که آنچه باید انجام شود، با بیانی سطح بالا و انتزاعی به رایانه داده شود و رایانه برنامه‌ای برای آن مسئله به یک زبان برنامه‌سازی بنویسد! قطعاً این مسئله سختی است چون دانش و جزئیات حل و پیاده‌سازی مسئله‌ها در ذهن انسان‌هاست، خودکارسازی این کار هوشمندی خاصی نیاز دارد. یک راه برای نیل به این هدف، جست‌وجو در فضای حالت همه برنامه‌های ممکن است تا برنامه‌ای مناسب تعریف مسئله پیدا شود. مشکل این راه‌حل اینست که فضای حالت همه برنامه‌های ممکن تقریباً بی‌نهایت است و جست‌وجوی کورکورانه در آن غیرعملی است. برای مقابله با این مشکل محققان به فکر استفاده از فنون و الگوریتم‌های فرا‌ابتکاری<sup>۱</sup> افتادند. یک شاخه از الگوریتم‌های فرا‌ابتکاری، الگوریتم‌های مُلهم از طبیعت<sup>۲</sup> هستند و الگوریتم‌های هوش توده‌ای (ازدحامی)<sup>۳</sup> هم زیرمجموعه‌ای از این الگوریتم‌ها هستند.

الگوریتم‌های هوش توده‌ای برای جست‌وجو در فضای حالت بسیار مناسبند و خواص ویژه‌ای دارند که آن‌ها را برای حل مسئله برنامه‌سازی خودکار<sup>۴</sup> و فائق آمدن بر مشکلات آن مناسب می‌سازد. این الگوریتم‌ها هوشمندانه در فضای حالت جست‌وجو می‌کنند که جواب‌های بهتر پیدا شود؛ این الگوریتم‌ها محدودیت‌های بسیاری حین جست‌وجو در نظر می‌گیرند که سریعتر به جواب برسند. با توجه به مطالب ذکر شده، این مقاله قصد دارد فنونی را از هوش توده‌ای که برای حل مسئله برنامه‌سازی خودکار استفاده شده برشمارد و به‌طور خلاصه آن‌ها را مرور نماید. این دسته از فنون

---

<sup>1</sup> Metaheuristic

<sup>2</sup> Bio-inspired

<sup>3</sup> Swarm intelligence

<sup>4</sup> Automatic programming

را می‌توان «برنامه‌سازی توده‌ای<sup>۱</sup>» نامید.

## ۱-۲- الگوریتم‌های ملهم از طبیعت، فرا ابتکاری و هوش توده‌ای

الگوریتم‌های الهام گرفته از طبیعت بر اساس سامانه‌های زیستی طراحی می‌شوند که خصوصیت‌های این سامانه‌های طبیعی را تقلید می‌نمایند. این سامانه‌ها و نظام‌های طبیعی ساختارهای نسبتاً ساده‌ای دارند، از مؤلفه‌های تصادفی استفاده می‌کنند، پیاده‌سازی رایانه‌ای ساده‌ای دارند و متنوع هستند. این ویژگی‌ها باعث جذابیت آن‌ها به‌عنوان روشی خوب برای طراحی الگوریتم می‌گردد. اغلب این الگوریتم‌ها برای حل مسائل بهینه‌سازی مورد استفاده قرار می‌گیرند که اصل مسئله NP سخت است و با این الگوریتم‌ها به‌دنبال به‌دست آوردن جواب‌های نزدیک به بهینه هستیم. شبکه عصبی مصنوعی<sup>۲</sup>، الگوریتم‌های تکاملی<sup>۳</sup>، سامانه‌های ایمنی مصنوعی<sup>۴</sup> و هوش توده‌ای از انواع مهم الگوریتم‌های ملهم از طبیعت هستند.

اغلب الگوریتم‌های مذکور و بسیاری الگوریتم‌های دیگر متعلق به خانواده‌ای از الگوریتم‌های هوشمند هستند که آن‌ها را فرا ابتکاری می‌نامند. علت این نام‌گذاری اینست که این الگوریتم‌ها از چند بخش ابتکاری (مکاشفه‌ای)<sup>۵</sup> تشکیل می‌شوند و کل این اجزا را یک الگوریتم ابتکاری بزرگ‌تر کنترل می‌نماید و این الگوریتم بزرگ‌تر فرا ابتکاری نامیده می‌شود. برای نمونه الگوریتم ژنتیک مثال مشهور فرا ابتکاری‌هاست. در این الگوریتم، اجزای کوچکتری مثل عملیات برش<sup>۶</sup> و جهش<sup>۷</sup> داریم و نیز تابعی موسوم به برازندگی<sup>۸</sup> که طراحی همگی ابتکاری است. در سطح بالاتر، طرز و دفعات و توالی استفاده از عملیات برش و جهش و محاسبه برازندگی را الگوریتم ژنتیک کنترل می‌کند و از این‌رو فرا ابتکاری نامیده می‌شود.

هوش توده‌ای خانواده‌ایست از الگوریتم‌ها با ویژگی‌های مشترک که همگی ملهم از طبیعت هستند و همگی فرا ابتکاری‌اند. در این الگوریتم‌ها جمعیتی از عامل‌های ساده وجود دارند که هر عامل رفتار مستقل و تصادفی از خود نشان می‌دهد. هر عامل با محیط و سایر عامل‌ها تعامل می‌کند و

---

<sup>1</sup> Swarm programming

<sup>2</sup> Artificial Neural Network (ANN)

<sup>3</sup> Evolutionary Algorithm (EA)

<sup>4</sup> Artificial Immune System (AIS)

<sup>5</sup> Heuristic

<sup>6</sup> Crossover

<sup>7</sup> Mutation

<sup>8</sup> Fitness

اطلاعات منتقل می‌نماید. کل سامانه به‌صورت نامتمرکز و خود تنظیمی و بدون نظارت و کنترل مرکزی کار می‌کند. عملکرد همه عامل‌ها باعث ظهور رفتار هماهنگ و هوش سراسری می‌گردد. این سامانه‌ها بسیار انعطاف‌پذیرند و در حل مسایل پیچیده و متنوع کارایی قابل توجهی از خود نشان می‌دهند. به‌کارگیری جمعیت زیاد باعث مقیاس‌پذیری، استحکام و وفق‌پذیری این الگوریتم‌ها می‌شود. بهینه‌سازی کلونی مورچه<sup>۱</sup>، بهینه‌سازی توده‌ای جزئی (ذره‌ای)<sup>۲</sup>، کلونی زنبورهای مصنوعی<sup>۳</sup>، بهینه‌سازی توده ماهی‌های مصنوعی<sup>۴</sup>، الگوریتم گله‌ای<sup>۵</sup> و کرم شب‌تاب<sup>۶</sup> از نمونه‌های مشهور الگوریتم‌های هوش توده‌ای هستند.

به گفته میلوناس<sup>۷</sup>، همه توده‌ها هوشمند نیستند. بنابراین او پنج خصوصیت را شناسایی کرد که هر توده باید داشته باشد تا هوشمند شناخته شود:

(الف) همجواری: توده باید قادر باشد محاسبات مکانی و زمانی را انجام دهد.

(ب) کیفیت: جمعیت توده باید بتواند به عوامل کیفی معینی پاسخ دهد؛ مثل میزان غذا، مکان و غیره.

(ج) پاسخ متنوع: منابع باید در بین شاخه‌های مختلف توزیع شوند تا در مقابل تغییرات محیط ایمن‌تر بمانند.

(د) پایداری و وفق‌پذیری: اولی یعنی توده نباید با تغییرات محیط رفتار متفاوتی نشان دهد، چون ممکن است قادر به جبران هزینه و انرژی از دست رفته نباشد؛ دومی یعنی اگر تغییر رفتار ارزش هزینه و انرژی مصرفی را داشته باشد، توده باید این کار را انجام دهد. این دو خصوصیت، دو روی یک سکه‌اند.

## ۱-۳- فرا ابتکاری‌های کلاسیک برنامه‌سازی خودکار

برنامه‌سازی ژنتیک<sup>۸</sup> و گونه‌های توسعه یافته آن همچون تکامل گرامری<sup>۹</sup> و برنامه‌سازی ژنتیک

---

<sup>1</sup> Ant Colony Optimization (ACO)

<sup>2</sup> Particle Swarm Optimization (PSO)

<sup>3</sup> Artificial Bee Colony (ABC)

<sup>4</sup> Artificial Fish Swarm Optimization (AFSO)

<sup>5</sup> Herd

<sup>6</sup> Firefly

<sup>7</sup> Millonas

<sup>8</sup> Genetic Programming (GP)

<sup>9</sup> Grammatical Evolution (GE)

هدایت شده با گرامر<sup>۱</sup> از روش‌های استفاده شده برای مسئله برنامه‌سازی خودکار هستند که بر مبنای هوش توده‌ای کار نمی‌کنند ولی شباهت‌های بسیاری دارند و بسیار رایجند. برنامه‌سازی ژنتیک الگوریتمی است فرا ابتکاری، تکاملی و ملهم از طبیعت که از الگوریتم ژنتیک برای عمل جست‌وجوی خود استفاده می‌کند ولی هر یک از اعضای جمعیت آن یک برنامه رایانه‌ای است که با نمایش خاصی مثل درخت اشتقاق<sup>۲</sup> کدگذاری<sup>۳</sup> شده است. در ابتدا تعداد مشخصی از افراد<sup>۴</sup> که هر فرد یک برنامه رایانه‌ایست به تصادف تولید می‌شود. سپس در تکرارهای الگوریتم ژنتیک، عملیات برش و جهش روی افراد اعمال می‌شود تا برنامه‌های جدیدتر تولید شود. مناسب بودن هر برنامه تولید شده با تابع برازندگی اندازه‌گیری می‌شود و تکرار تا برآورده شدن شرط خاصی ادامه پیدا می‌کند.

برنامه‌سازی ژنتیک ساده سه مشکل عمده دارد: یکی از دست رفتن مقدراری اطلاعات حین کدگذاری و عمل جست‌وجو؛ دوم مسئله تورم کد<sup>۵</sup> و سوم تولید افرادی (برنامه‌هایی) که از نظر گرامری نامعتبرند<sup>۶</sup>. برای حل این مشکل در گونه‌های توسعه یافته برنامه‌سازی ژنتیک از گرامر مستقل از متن<sup>۷</sup> زبان برنامه‌سازی استفاده شده است.

## ۱-۴- فرا ابتکاری‌های هوش توده‌ای

در بخش‌های قبل الگوریتم‌های فرا ابتکاری که تاکنون در برنامه‌سازی خودکار مورد استفاده قرار گرفته‌اند برشمرده شدند. در این بخش گونه‌ها و جزئیات بیشتری ارائه می‌شود. بررسی روند مقالات نشان می‌دهد بعد از برنامه‌سازی ژنتیک، رایج‌ترین فرا ابتکاری مورد استفاده برای برنامه‌سازی خودکار، استفاده از بهینه‌سازی کلونی مورچه‌ها بوده است و می‌توان آن‌ها را «برنامه‌سازی مورچه‌ای» نامید. دوریگو برای اولین بار ایده استفاده از رفتار مورچه‌ها را پیشنهاد کرد. بر این اساس رفتار یافتن غذا، ترشح فرومون برای انتقال اطلاعات و مسیریابی مورچه‌ها و سایر رفتار آن‌ها مبنای طراحی خانواده بزرگی از گونه‌های الگوریتم کلونی مورچه‌ها شد.

---

<sup>1</sup> Grammar-Guided GP (G3P)

<sup>2</sup> Parse tree

<sup>3</sup> Encode

<sup>4</sup> Individual

<sup>5</sup> Code bloat

<sup>6</sup> Syntactically invalid individuals

<sup>7</sup> Context Free Grammar (CFG)

در حوزه برنامه‌سازی خودکار، برنامه‌سازی کلونی مورچه<sup>۱</sup>، برنامه‌سازی کلونی مورچه مشبک<sup>۲</sup>، برنامه‌سازی مورچه‌ای پویا<sup>۳</sup>، برنامه‌سازی مورچه‌ای تعمیم‌یافته<sup>۴</sup>، برنامه‌سازی مورچه‌ای تعمیم‌یافته تقویت شده<sup>۵</sup>، برنامه‌سازی مورچه‌ای کارت‌زین<sup>۶</sup>، برنامه‌سازی مورچه‌ای مبتنی بر گرامر<sup>۷</sup> و برنامه‌سازی مورچه‌ای مبتنی بر گرامر<sup>۸</sup> چند هدفه از نمونه‌های مشهور هستند.

هوش توده‌ای جزئی در رتبه دوم بعد از کلونی مورچه قرار دارد که برای طراحی الگوریتم‌های برنامه‌سازی خودکار مورد استفاده قرار گرفته است. حرکت توده پرنده‌گان حین پرواز نمونه طبیعی چنین الگوریتم‌هایی است. معمولاً برای هر فرد (مثلاً پرنده) یک موقعیت مکانی و سرعت حرکت در نظر گرفته می‌شود و این داده‌ها در قالب بردارهایی در فضای هندسی نمایش داده می‌شود و طبق قوانین فرا ابتکاری اصلی هوش توده‌ای جزئی، الگوریتم عمل می‌کند. توده گرامری<sup>۹</sup>، بهینه‌سازی توده درخت‌ها<sup>۱۰</sup>، بهینه‌سازی توده‌ای جزئی هندسی<sup>۱۱</sup> و ترکیب برنامه‌سازی ژنتیک با بهینه‌سازی توده‌ای جزئی<sup>۱۲</sup> از گونه‌های مورد استفاده برای حل مسئله برنامه‌سازی خودکار بوده‌اند.

رتبه سوم الگوریتم‌ها به فرا ابتکاری‌های برنامه‌سازی توده زنبورها<sup>۱۳</sup> اختصاص دارد که از الگوریتم کلونی زنبورها<sup>۱۴</sup> برای جست‌وجو در فضای حالت برنامه‌های رایانه‌ای در مسئله برنامه‌سازی خودکار استفاده می‌کنند. زنبورها رفتارهایی همچون رقص و ارتباط، ازدواج، تخصیص وظایف، یافتن غذا و تصمیم گروهی دارند و این‌ها مبنای طراحی الگوریتم فرا ابتکاری است. کلونی زنبور گرامری<sup>۱۵</sup> یکی از گونه‌های این خانواده است.

سه فرا ابتکاری برنامه‌سازی توده ماهی‌های مصنوعی، برنامه‌سازی گله‌ای و گله گرامری و برنامه-

---

<sup>1</sup> Ant Colony Programming (ACP)

<sup>2</sup> Grid ACP (GACP)

<sup>3</sup> Dynamic Ant Programming (DAP)

<sup>4</sup> Generalized AP (GAP)

<sup>5</sup> Enhanced GAP (EGAP)

<sup>6</sup> Cartesian AP (CAP)

<sup>7</sup> Grammar-Based AP (GBAP)

<sup>8</sup> Multi-Objective GBAP (MOGBAP)

<sup>9</sup> Grammatical Swarm (GS)

<sup>10</sup> Tree Swarm Optimiziation (TSO)

<sup>11</sup> Geometric PSO (GPSO)

<sup>12</sup> Hybrid Genetic Programming with PSO (HGPPSO)

<sup>13</sup> Bee Swarm Programming (BSP)

<sup>14</sup> Bee Colony Algorithm (BCA)

<sup>15</sup> Grammatical Bee Colony (GBC)

سازی کرم شب‌تاب از سایر فرا ابتکاری‌هایی هستند که از سال ۲۰۱۲ به بعد در حوزه برنامه‌سازی خودکار مورد استفاده قرار گرفته‌اند.

## ۱-۵- جمع‌بندی

استفاده از فرا ابتکاری‌های برگرفته از طبیعت روش جست‌وجوی مناسبی برای غلبه بر مشکلات فضای حالت در بسیاری مسایل کاربردی به‌خصوص برنامه‌سازی خودکار است. مقاله حاضر صرفاً برخی از مفاهیم لازم را معرفی کرد و با برشمردن الگوریتم‌های فرا ابتکاری مورد استفاده برای برنامه‌سازی خودکار آن‌ها را به جویندگان زمینه‌های تحقیقی معرفی کرد.

## مراجع

- [1] Sipser, Michael. *Introduction to the Theory of Computation*. 3<sup>rd</sup> Edition. Boston: Thomson Course Technology, 2013.
- [2] Dorigo, Marco, Thomas, Stutzle. *Ant colony optimization*. The MIT Press, Cambridge, Massachusetts, 2004.
- [3] Kennedy, James F., James Kennedy, Russell C. Eberhart, and Yuhui Shi. *Swarm intelligence*. Morgan Kaufmann, 2001.
- [4] Floreano, Dario, and Claudio Mattiussi. *Bio-inspired artificial intelligence: theories, methods, and technologies*. MIT press, 2008.
- [5] Yang, Xin-She, Zhihua Cui, Renbin Xiao, Amir Hossein Gandomi, and Mehmet Karamanoglu, eds. *Swarm intelligence and bio-inspired computation: theory and applications*. Newnes, 2013.
- [6] Olmo, Juan L., José R. Romero, and Sebastián Ventura. "Swarm-based metaheuristics in automatic programming: a survey." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4, no. 6 (2014): 445-469.