معرفی الگوریتم های فرهنگی و ارائه یک پیاده سازی جدید

با موضوع تکامل ذائقه غذایی افراد به هنگام حضور در جوامع اجتماعی

Introduce and Utilizition of Cultural Algorithms

with Implementation of a new Algorithm

پروژه پایانی دوره کارشناسی

**استاد راهنما : سرکارخانم دکتر مینا زلفی**

علی فاضـــــــلی

دانشکده برق و مهندسی کامپیوتر دانشگاه تبریز

تابستان 96

Ç

**فهرست مطالب**

[مقدمه : 4](#_Toc490562644)

[قسمت های مختلف الگوریتم فرهنگی: 5](#_Toc490562645)

[روش‌های تغییرات جمعیت در الگوریتم فرهنگی: 7](#_Toc490562646)

[ویژگی های اساسی الگوریتم فرهنگی: 7](#_Toc490562647)

[زمینه‌های مناسب برای بکارگیری الگوریتم فرهنگی: 9](#_Toc490562648)

[ساختار کلی پروژه:……. 10](#_Toc490562649)

[تاریخچه: 10](#_Toc490562650)

[پیشینه استفاده از مدل محاسباتی تکامل فرهنگی: 11](#_Toc490562651)

[مولفه‌های دانش الگوریتم فرهنگی: 13](#_Toc490562652)

[مولفه وضعی (Situational Component) : 13](#_Toc490562653)

[مولفه هنجاری (Normative Component) : 13](#_Toc490562654)

[مولفه دامنه اسمی دانش (Domain Knowledge) : 14](#_Toc490562655)

[مولفه‌های گزینش جمعیت در الگوریتم فرهنگی: 14](#_Toc490562656)

[تابع پذیرش (Acceptance Function) : 14](#_Toc490562657)

[بروز رسانی فضای اعتقادی ( فرهنگ) : 1- مولفه وضعی : 15](#_Toc490562658)

[تابع تاثیر(Influence Function) : 17](#_Toc490562659)

[صرفا استفاده از مولفه هنجاری : 17](#_Toc490562660)

[صرفا استفاده از مولفه وضعی برای تشخیص جهت حرکت: 17](#_Toc490562661)

[استفاده از مولفه موضعی برای تعیین جهت و مولفه هنجاری برای میزان حرکت: 18](#_Toc490562662)

[استفاده از مولفه هنجاری هم برای تعیین جهت و هم برای تعیین میزان گام: 18](#_Toc490562663)

[مراحل (روند اجرا) الگوریتم فرهنگی: 19](#_Toc490562664)

[پیاده سازی اولیه الگوریتم فرهنگی: 20](#_Toc490562665)

[نیازمندی های پیاده سازی: 20](#_Toc490562666)

[قسمت اول – تعریف مساله : 21](#_Toc490562667)

[قسمت دوم – تنظیمات مربوط به الگوریتم فرهنگی: 22](#_Toc490562668)

[قسمت سوم – آماده سازی : 23](#_Toc490562669)

[قسمت چهارم - بروزرسانی بهترین پاسخ یافت شده: 24](#_Toc490562670)

[قسمت پنجم – آماده سازی فرهنگ: 25](#_Toc490562671)

[علت وجود اسکریپت AdjustCulture.m در الگوریتم چیست؟ 26](#_Toc490562672)

[اجرای الگوریتم فرهنگی در بستر نرم افزار متلب 28](#_Toc490562673)

[پیاده سازی شبیه ساز تکامل دستور‌های پخت غذا : 29](#_Toc490562674)

[طرح مساله:……………… 29](#_Toc490562675)

[مراحل انجام الگوریتم فرهنگی بر روی دستور‌های پخت: 31](#_Toc490562676)

[نحوه اجرای الگوریتم:….. 32](#_Toc490562677)

[نتیجه گیری و کارهای آینده: 34](#_Toc490562678)

[منابع: 35](#_Toc490562679)

# مقدمه :

الگوریتم‌های ژنتیک از گونه الگوریتم‌های تکاملی می‌باشند که با استفاده از ایده نحوه تاثیر جهش ژنتیکی و بهبود آن بر روی اعضای جامعه و با نگاه ژنتیکی و طبیعی به موضوع تکامل انسان‌ها در طبیعت توانسته خیل عظیمی از مسائل انسان را به خوبی مدل کند. اما در الگوریتم های فرهنگی فرایند تکامل افراد جامعه بصورت دوگانه، هم از جهت تکامل ژنتیکی برخواسته از طبیعت انسان، و هم با استفاده از فرهنگ موجود در جامعه برخواسته از تمدن آن‌ها انجام می‌گیرد. منطقا غربال‌گری و تکامل جمعیت با استفاده از دو فاکتور نتیجه بسیار مطلوب‌تری در بسیاری از مواقع بدست خواهد داد. که اینگونه الگوریتم‌ها را "الگوریتم فرهنگی" و یا "الگوریتم تکاملی فرهنگی" می‌نامند.

این الگوریتم‌ها موارد مشابه یا اصطلاحا Counter Part های الگوریتم ژنتیک می‌باشند. همانطور که در الگوریتم ژنتیک شاهد حل مسائل با استفاده از تکامل زیستی هستیم در الگوریتم‌های فرهنگی نیز به دنبال استفاده از تکامل فرهنگی یک جامعه، فضای فرهنگی حاکم بر آن و تعامل و یا تقابل فرهنگ در برابر جامعه به منظور حل مسائل می‌باشیم. در اینجا منظور از جامعه تاحدود بسیار زیادی با مجموعه ژنوم‌ها در الگوریتم ژنتیک مشابهت دارد و این المان می‌تواند به عنوان هر پارامتری ایفای نقش کند. این مهم با استفاده از پیاده‌سازی و ساخت یک مدل انجام می‌گیرد که در نهایت موجب کمک به حل مسائل بهینه‌سازی خواهد شد.

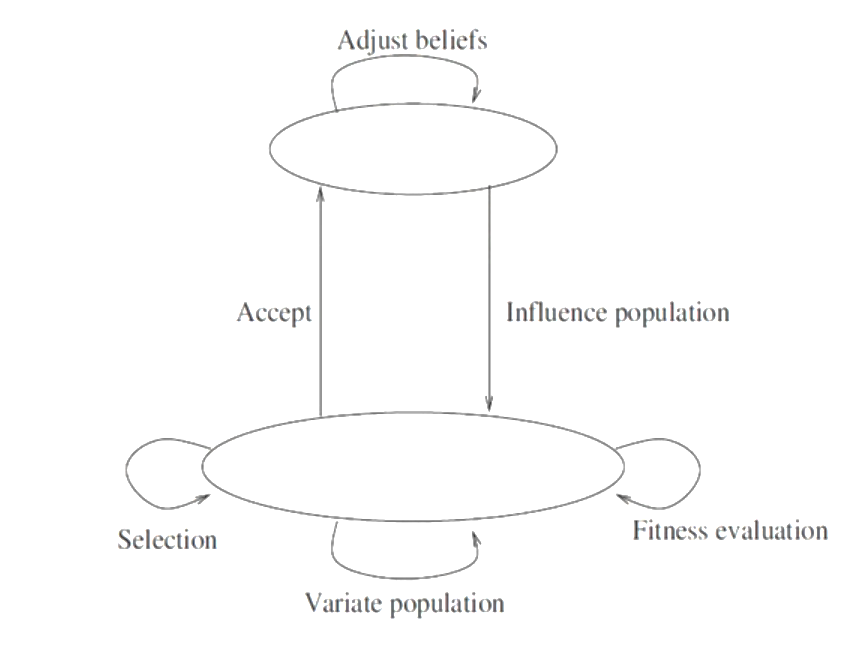
## قسمت های مختلف الگوریتم فرهنگی:

در ابتدا جالب است که بحث را با چند سوال آغاز کنیم:

سوالی که در اینجا مطرح می‌شود این است که تکامل چگونه انجام می‌پذیرد؟ در یک اجتماع چه افرادی نقش بیشتری در تولید و تغییر فرهنگ دارند؟ الگوریتم فرهنگی پاسخ این سوال را در وجود "نخبگان جامعه" می داند. نخبه کسی است که در جامعه وزن و تاثیر بیشتری داشته باشد و یا به عبارتی بیشتر مورد توجه باشد. این گونه افراد "الگو" نیز نامیده می شوند. این فرد میتواند یک بازیگر یا یک سیاست مدار یا یک ورزشکار باشد که با استفاده از تمکن مالی خود و یا شهرت و یا قدرت نفوذ در اجتماع خود به اعمال سلیقه شخصی خود در رفتار عمومی جامعه بپردازد. طبیعتا رفتار و جهت گیری‌های یک فرد مشهور و مورد توجه در جامعه بر روی جامعه تاثیر بسزایی داشته و آرام آرام رفتارش تبدیل به فرهنگ در جامعه می شود. این اتفاق با عنوان "تاثیر افراد بر روی فرهنگ" و "ایجاد مولفه فرهنگی" بررسی می شود.

اما در رابطه با "تاثیر فرهنگ بروی افراد" این توضیح وجود دارد که هنگامیکه رشته ای از افراد یک مولفه فرهنگی را در جامعه ایجاد می‌کنند. و یک فرهنگ بر جامعه غالب می شود، دسته ای از مردم به دلیل آنکه محل ارجاع رفتار‌های اجتماعی‌شان فرهنگ می‌باشد، از این مولفه فرهنگی (فرهنگ) تاثیر پذیرفته و فرهنگ پیاده‌سازی می‌شود. و پس از آن افراد با اعمال سلیقه شخصی خود قسمت کوچکی از آن مولفه فرهنگی را با تغییرات کوچک شخصی‌سازی کرده و از آن پیروی می‌کنند.

فرض کنید یک فرد در یک جامعه یک مولفه فرهنگی جدید (مانند استفاده از کتب امانی در وسایل نقلیه عمومی) را می‌پذیرد. اما طبیعتا هر فرد مقداری از سلیقه شخصی خود را در فرهنگ غالب جامعه دخیل می‌کند. بطور مثال این فرد یک تکه سفید برداشته، بخش‌های جالب کتاب عمومی را در آن یادداشت کرده و دوباره لای کتاب می‌گذارد. این مولفه فرهنگی شخصی‌سازی شده (در موارد مثبت می گوییم بهبود یافته) در جامعه پذیرفته می‌شود و هر فرد دیگری که این مولفه فرهنگی را می‌پذیرد(استفاده از کتب عمومی در وسایل نقلیه) عمل نکته برداری نکات جالب کتاب را نیز انجام می‌دهد. و این‌گونه است که بهبود یک مولفه فرهنگی انجام می‌پذیرد. در کل اساس کار الگوریتم‌های فرهنگی همین نکته می‌باشد.



تصویر 1 ساختار کلی یک الگوریتم فرهنگی

در شکل بالا قسمت بالایی شکل نشان دهنده فرهنگ و قسمت پایین نشان دهنده جمعیت می‌باشد. ما یک بده بستان بین فرهنگ و جمعیت داریم. تاثیری که جمعیت بر فرهنگ دارد اصطلاحا اثر acceptance یا مقبولیت می‌باشد ( همرنگ جماعت شو). و بطور متقابل تاثیری که فرهنگ بر روی جمعیت مورد مطالعه دارد را تاثیر یا influence گویند.

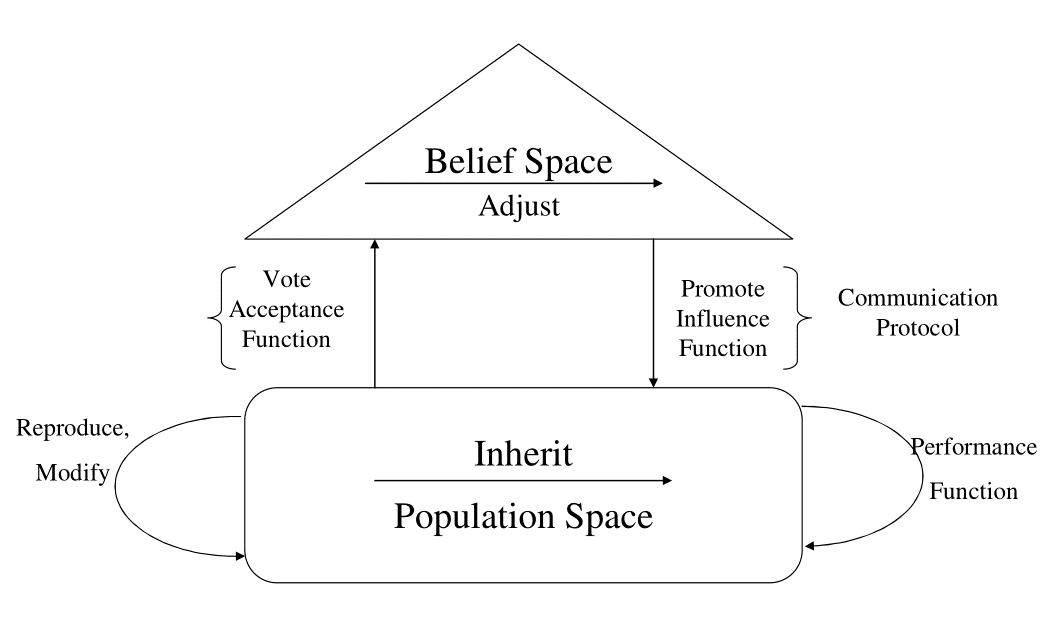
بطور مثال، افرادی که وزن بیشتری از لحاظ ارزشی در جامعه دارند ( به عبارتی نخبه هستند) ، نرخ مقبولیت بیشتری نسبت به دیگر اعضای جامعه دارند.

تغییراتی که در فرهنگ انجام می‌پذیرد و تاثیراتی که فرهنگ بر روی فرهنگ می‌گذارد را Adjust Belief می گویند. گاها فرهنگ جدیدی بوجود می آید که فرهنگ‌های دیگر را تحت تاثیر قرار می دهد.

# روش‌های تغییرات جمعیت در الگوریتم فرهنگی:

از جمله تغییراتی که در جمعیت رخ میدهد می توان Selection ، Variate Population و Fitness Evaluation را نام برد که در ادامه به هر‌کدام از آنان اشاره خواهیم‌کرد:

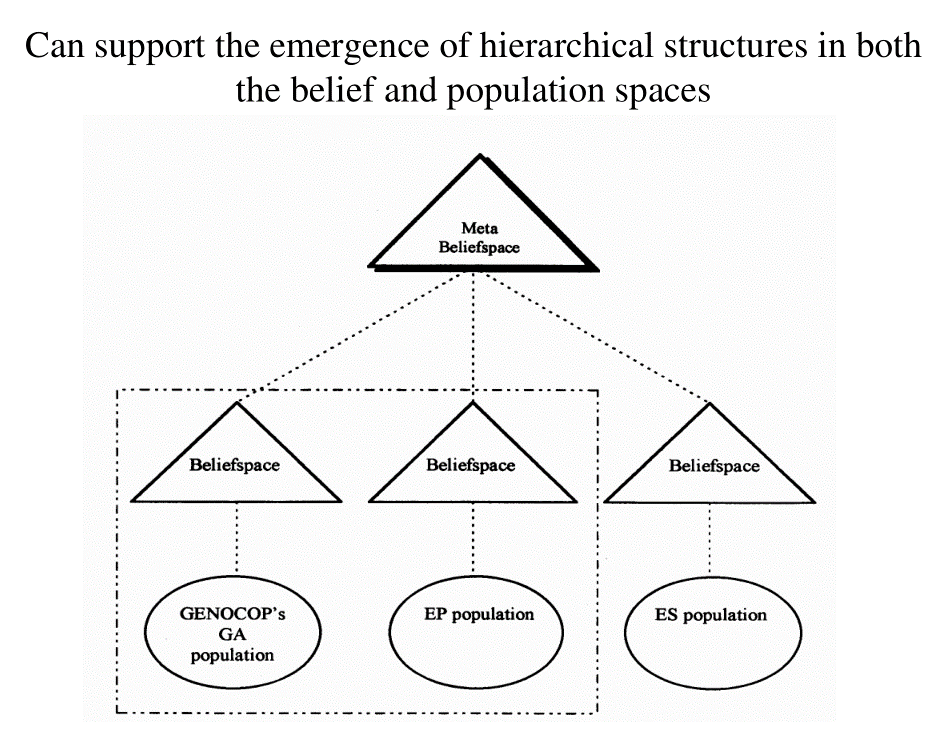
* Selection یا انتخاب : فرآیند وزن دهی به عناصر جامعه را گویند. اتخاذ فرایندی که طی آن عناصر مهم جامعه از دیگر عناصر متمایز شوند (و متعاقبا تاثیر بیشتری بر روی فرهنگ داشته باشند) را انتخاب گویند.
* Fitness Evaluation: به معنای ارزیابی جمعیت بهتر می باشد.
* و در آخر قسمت Variate Population که به معنای تغییر اعضای جمعیت می باشد که در آن شیوه هایی همچون mutation و یا crossover انجام پذیرد. این Variation ها در الگوریتم ژنتیک کلاسک نیز وجود دارد.اما دخالت دادن فرهنگ و مقادیر بهینه در فرایند اصلاح رویکردی است که در الگوریتم های فرهنگی بدان پرداخته می شود.



تصویر 2 اشکال مختلف ارتباطات بین اعضای الگوریتم فرهنگی

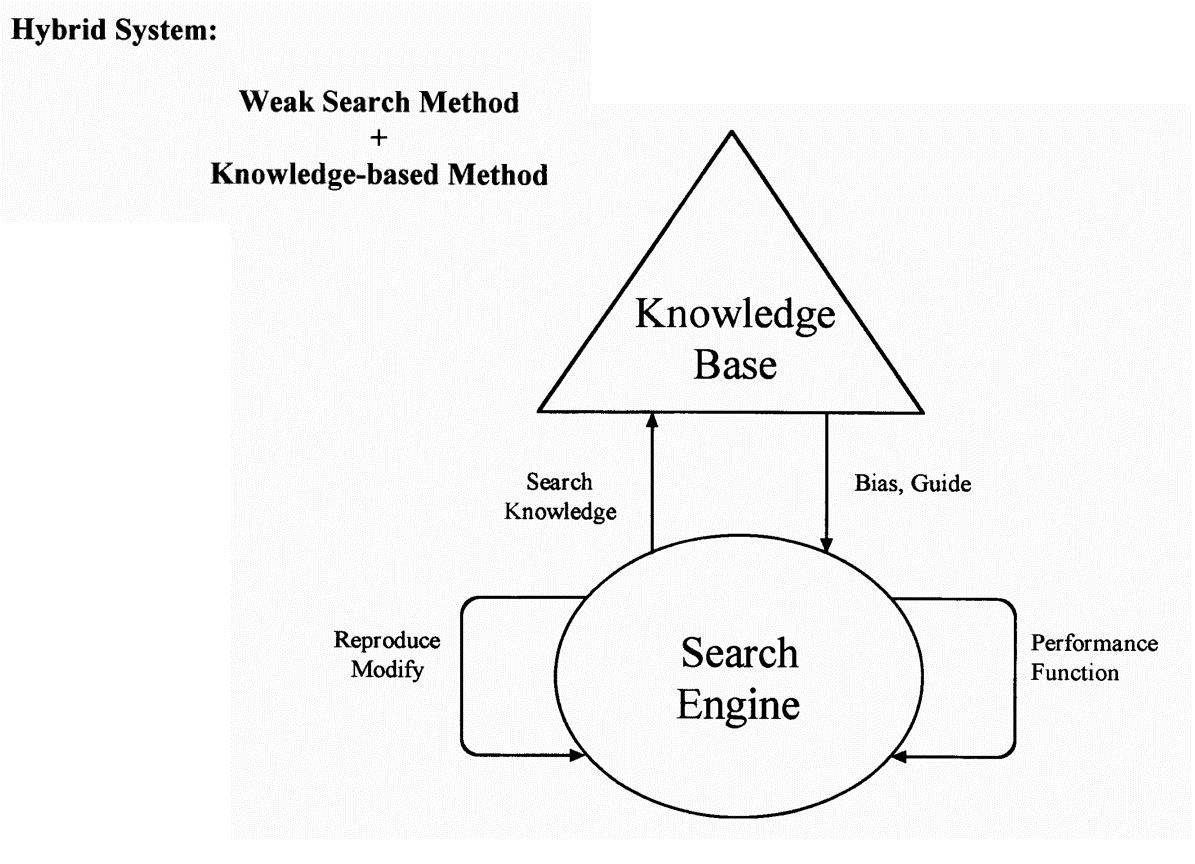
## ویژگی های اساسی الگوریتم فرهنگی:

* تاثیر پذیری و ارث بری دوگانه، از طریق جمعیت و سطوح دانش (در این الگوریتم بالاخص فرهنگ)
* دانش یا همان فرهنگ به عنوان چراغ راه مسیردهی به اعضای جمعیت به منظور بهبود روند جست و جو می باشد
* قابلیت پشتیبانی از طراحی روال جست و جوی سلسله مراتبی براساس جمعیت و یا فضای اعتقادی(فرهنگ)



تصویر 3 نحوه ارتباطات سلسله مراتبی بین اعضای الگوریتم فرهنگی

* جداسازی دانش برگزیده (فرهنگ) از دانش فردی (فرهنگ هریک از اعضای جمعیت)
* پشتیبانی از انطباق خودکار مولفه فرهنگی در سطوح مختلف
* امکان انجام تکامل فرهنگی با نرخ ها و سطوح مختلف
* پشتیبانی از رویکرد های ترکیبی (hybrid) برای حل مسائل



تصویر 4 پشتیبانی از رویکرد های ترکیبی برای حل مسائل

## زمینه‌های مناسب برای بکارگیری الگوریتم فرهنگی:

ویژگی‌های ذاتی الگوریتم فرهنگی (که در بالا به آن اشاره شد) باعث می‌شود که این الگوریتم پتانسیل بالایی برای حل مسائل زیر داشته باشد:

* مسائلی که در آن‌ها مقادیر domain knowledge قابل توجه و زیاد باشد. بطور مثال در مسائل بهینه سازی مطلق(اجباری) (Constrained optimization problems)
* مسائل پیچیده مربوط به سیستم هاییکه که عمل انطباق در سطوح و نرخ های مختلف در جمعیت و یا فضای اعتقادی صورت می گیرد.
* دانش در دسترسی دارای حالات و اشکال مختلف باشد و با معیار‌ها و رویکرد‌های متفاوت مورد تصمیم گیری واقع شود
* سیستم های ترکیبی(hybrid) ای که نیاز به ترکیبی از جست و جوی صرف و چهارچوب‌های برپایه دانش دارند.
* حل مسائلی که نیازمند چندین گروه مختلف جمعیت و فضای اعتقادی، و متعاقبا ارتباط بین آن‌ها دارد.
* مسائل دارای ساختار سلسله مراتبی که ساختار جمعیت و یا روند تاثیرپذیری یا تاثیرگذاری آنها بروی فرهنگ بصورت سلسله مراتبی مدل می شوند.

## ساختار کلی پروژه:

در ابتدا مقدمات و مفاهیم مورد نیاز برای درک الگوریتم فرهنگی مطرح شد، سپس به تاریخچه مختصری از نحوه پیداش این نوع الگوریتم پرداخته میشود. سپس الگوریتم اولیه در نرم افزار متلب پیاده سازی و شرح داده می‌شود تا از تمامی جوانب و راه حل‌های موجود برای مواجهه با مساله با استفاده از الگوریتم فرهنگی با خبر شویم. در آخر مدل پیشنهادی اینجانب که استفاده کاربردی از الگوریتم‌های فرهنگی به منظور کاربردی‌سازی یک مفهوم نظری می‌باشد را شاهد هستیم.

# تاریخچه:

در سال 1881 فردی بنام ادوراد ب. تیلور اولین دانشمند و جامعه شناسی بود که از لفظ "فرهنگ" استفاده نمود. او در کتابش بنام "فرهنگ اولیه" در همان سال فرهنگ را بصورت زیر تعریف کرد:

"مجموعه‌ای مرکب از دانش، اعتقادات، هنر، اخلاقیات، رسوم و دیگر قابلیت‌ها و عادت‌های بشر به عنوان جزئی از جامعه متمدن."

در سال 1957 جورج مورداچ کتابی منتشر نمود که 565 فرهنگ مختلف بر اساس شخصیت 30 نفر ، مورد بررسی قرار گرفته بود.

در سال 1960 علم بررسی عادات و نحوه زندگی (اصطلاحا Cultural Ecology) بصورت جدی به شکل تعاملات بین نظام فرهنگی جامعه و محیط بررسی شدند.

برخی از تعاریف محققان در این زمینه نیز در مطالعات بسیار حائز اهمیت است. که از این جمله تعاریف میتوان به تعریف گریتز در سال 1973 و تعریف دورهام در سال 1990 اشاره کرد:

"فرهنگ معنای مطلق تفسیری است که انسان‌ها از تجربیات و رفتار خود دارند." گریتز 1973

"فرهنگ به معنای به اشتراک گذاشتن پدیده‌های مشترک اتفاق افتاده در برابر انسان‌ها می باشد" دورهام 1990

سوال هاییکه در طراحی یک الگوریتم فرهنگی پیش می‌آید این است که :

* چگونه برای یک عضو مفروض جامعه عملیات مقبول شدن (acceptance) انجام می‌پذیرد. و چگونه در تغییر فرهنگ نقش دارد.
* اعضای جمعیت را با هدف تاثیرگذاری در فرهنگ چگونه انتخاب کنیم؟ سیاست‌گذاری ما برای انتخاب افراد در جمعیت چگونه باید باشد؟
* چگونه تغییرات فرهنگی را بر روی جمعیت اعمال کنیم؟

در الگوریتم فرهنگی قسمت فرهنگ یا Culture نام‌گذاری کلی‌ای است که بصورت دقیق‌تر و استاندارد‌سازی شده در الگوریتم‌های تکاملی با نام فضای اعتقادی یا Belief Space خوانده می‌شود. که ویژگی‌های توصیفی و توصیف کنندگی آن بیشتر مد نظر می‌باشد. لذا ما یک فضای اعتقادی داریم که حاوی یک سری مولفه‌های دانش (Knowledge Components) می باشد.

# پیشینه استفاده از مدل محاسباتی تکامل فرهنگی:

پیشینه مدل سازی محاسباتی بر اساس تکامل فرهنگی به سال 1980 بر می گردد که کاوالی اسفروزا و فلدمن (Cavalli-Sforza & Feldman, 1981) اولین محاسبات کمی وابسته به یک مدل را با استفاده یک الگوریتم فرهنگی ارائه دادند.که این امر با استناد بر اسناد ایشان و با استفاده از تحقیقات بسیار گسترده در زمینه اجتماع و روانشناسی ( و به خصوص تکامل در زمینه تکامل زبان) صورت گرفته بوده است.

Cavalli-Sforza و Feldman مدل‌هایی از نحوه برخورد و تعامل فرهنگی انسان ها استخراج و تولید کردند که نه تنها تکامل با استفاده از انتقال هنجارها بوسیله والدین بیولوژیکی افراد (یادگیری اجتماعی عمودی) صورت می گرفت، بلکه این امر از هم سطح های یک عامل نیز صورت می‌گرفته است (یادگیری اجتماعی افقی) و حتی با استفاده از اعضای نامربوط از عوامل هم سطح والدین عامل و یا نیاکان آنها ( که اصطلاحا یادگیری اجتماعی اریب نامیده می شود.).

آنها مدل‌های جهش فرهنگی را همانند جهش (mutation) ژنتیکی که از قبل موجود بود پیاده سازی کردند. درحالیکه رفتار‌های اجتماعی روز دهه 80 میلادی بیشتر با الگوی تصادفی مدل می شدند، اسفروزا و فلدمن غربال عوامل را با استفاده از انتخاب طبیعت (هر عاملی که دارای قدرت و میزان سازگاری با محیط بیشتری است باقی مانده و پیروز می شود) پایه‌گذاری کردند. بدین نحو که هنجار اجتماعی بر اساس شانس بقایی که در میان جامعه دارد می‌تواند تغییر کند.

در سال 1988 بوید و ریچاردسون روانشاسی واقع گرایانه را وارد الگوریتم تکاملی فرهنگی کردند. این امر با استناد بر اینکه عوامل مورد بررسی در الگوریتم (در اینجا مردم) دارای این توانایی هستند که با قائل شدن امتیازی برای خود از رفتار و هنجار افراد با امتیاز بالاتر از خود (شان اجتماعی بالاتر، همانند الگو ها و افراد شناخته شده جامعه) تقلید کنند.

این تقلید با استفاده از معیار‌های مختلفی صورت می‌گیرد. روش اول که در بالا به آن اشاره شد روش معیارغیر مسیقیم (Prestige bias) می باشد. روش معیارگزینی بعدی روشی است که در آن عوامل هنجار‌های اجتماعی را بر اساس محبوبیت عوامل برای خود بر می‌گزینند (این روش، مستقل از فراوانی و یا انطباق نیز نامیده می شود). روش معیار گزینی بعدی روشی است که انتخاب هنجار با استفاده از مشخصات و کاراکتر ذاتی عوامل صورت می گیرد. بدین صورت که که برخی ویژگی‌های خاص برخی عوامل مهم تعمدا برجسته شده و عوامل جامعه مجبور به تقلید از آن می شوند. از این دسته از معیار گزینی میتوان به قوانین اساسی جامعه و خط قرمز‌های آن اشاره کرد.

بوید و ریچاردسون همچنین مدلی ارائه کردند که در آن دو مکانیزم جداگانه برای انتقال و پذیرفتن هنجارها برای عوامل جامعه ارائه شد:

1. جهش فرهنگی تصادفی که بر اساس مدل قبلی اسفروزا و فلدمن در سال 1980 ارائه شده بود پایه گذاری شد.
2. دگرگونی تعلیمی (guided variation) : که در این مدل افراد جامعه اطلاعات لازمه برای رسیدن به اهداف غایی و یا تمایلات خود را خودشان بدست می آوردند.

در مدل ارائه شده در این مقاله، تمامی نقطه نظر های مختلف این اصول پیاده سازی شده است: Variation ، Selection و Replication.

برای ساده‌سازی هدف، دگرگونی(Variation) فقط مقادیر ورودی خود را از جهش فرهنگی بصورت تصادفی دریافت می‌کند. و فرایند انتخاب (Selection) براساس فرایند انطباق (conformity bias) صورت می گیرد. به عبارت ساده تر انتقال یک هنجار اجتماعی به کل اعضای جامعه در ابتدا باید در یک گروه اجتماعی خاص امتحان خود را پس بدهد، سپس این انتقال هنجار انجام گیرد. جزئیات بیشتر در قسمت های بعدی ارائه می شوند.

# مولفه‌های دانش الگوریتم فرهنگی:

## مولفه وضعی (Situational Component) :

از مهم‌ترین مولفه‌های الگویتم فرهنگی می‌توان به "مولفه وضعی" یا Situational Component اشاره کرد که آمار ویژگی‌های شاخص در فرهنگ را مشخص می کند. بطور مثال باهوش‌ترین فرد درون جمعیت یا مشهورترین عنصر جامعه و امثال این گونه ویژگی‌ها.

در الگوریتم‌های تکاملی پیشین همواره مقادیر اکسترمم مطلق (حداقل‌ها و حداکثر‌ها) متمایز شده و از دیگر اعضای جمعیت بگونه‌ای (باتوجه به نحوه پیاده‌سازی) جدا می‌شدند. اما در الگوریتم‌های فرهنگی علاوه بر نگهداری این مقادیر مقادیر بهینه در هرمرحله نیز نگهداری شده و از آن‌ها به منظور بهبود کارکرد الگوریتم استفاده می‌شود.

یا به عبارت دیگر اگر به الگوریتم ژنتیک کلاسیک یک فضای فرهنگی برای تصمیم گیری‌ها و تغییرات اضافه کنیم الگوریتم فرهنگی را خواهیم داشت. بطور مثال در الگوریتم PSO به جای اینکه global best و personal best داشته باشیم، میتوان نقطه‌ای را به عنوان هدف در نظر گرفت که فرهنگ پیشنهاد می‌کند.

## مولفه هنجاری (Normative Component) :

بطورمثال این مولفه "بازه تحمل رانندگان در زمان حرکت اتومبیل جلویی خود" را مدل می‌کند. بطور مثال به ازای سرعت 15km/h بایستی یک خودرو به اندازه طول خودرو خود از خودرو مقابل خود فاصله داشته باشد. و بطور مثال در سرعت 60km/h بایستی این فاصله به اندازه چهار برابر طول خودرو باشد. درصورتیکه جوامعی وجود دارند که رعایت این قانون نانوشته به ضرر راننده و سوء استفاده دیگر رانندگان منجر می شود. حال چه خوب یا بد این هنجار برای یک فرهنگ غالب جامعه می‌باشد. و طبیعتا مقادیر کمی یا کیفی این هنجار می‌تواند به عنوان یک مولفه هنجاری در الگوریتم فرهنگی مورد استفاده قرار گیرد.

## مولفه دامنه اسمی دانش (Domain Knowledge) :

که بسیار شبیه به مولفه وضعی می باشد. بدین صورت تعریف می‌شود که در هر تکرار و هر نسل از جمعیت مقادیر بهینه گزارش شده و در این مولفه نقش دارد. به عبارت دیگر هر هنگام که مقادیر اکسترمم در هر نسل را ذخیره کرده و از از آن‌ها برای بهبود استفاده کنیم می‌گوییم از مولفه دامنه اسمی دانش استفاده کرده ایم.

# مولفه‌های گزینش جمعیت در الگوریتم فرهنگی:

## تابع پذیرش (Acceptance Function) :

عبارت است از انتخاب یک عضو از جمعیت به منظور تاثیر‌گذاری بر روی فرهنگ و انتخاب اینکه چه افرادی به چه نحوی می‌توانند بر روی فرهنگ تاثیر بگذارند. تعداد افراد تاثیر گذار بر روی فرهنگ به دو روش ایستا و پویا می‌تواند مدیریت شود. به شکل ایستا همانطور که از نام بر می آید اینگونه است که تعداد افرادی مشخص از تمام جمعیت به عنوان افراد بهینه متمایز شده و از آنها برای تاثیرگذاری در فضای فرهنگی مورد استفاده قرار می گیرد. و در حالت پویا تعداد افراد موثر ثابت نیست و با استفاده از الگوریتم هایی کمیت و کیفیت این تاثیر از اعضای جمعیت متغیر خواهد بود.

البته مولفه‌های دیگری بصورت ابتکاری ابداع شده اند که تعداد آنان بسیار بوده و از حوصله این بحث خارج است.

# بروز رسانی فضای اعتقادی ( فرهنگ) : 1- مولفه وضعی :

همانطور که پیشتر هم اشاره شد مولفه وضعی یک رکود از بهترین پاسخ‌های فعلی تولید شده در مساله می باشد. پس این مولفه بهترین پاسخ تمام نسل ها و یا نسل جاری می باشد.

2- مولفه هنجاری :

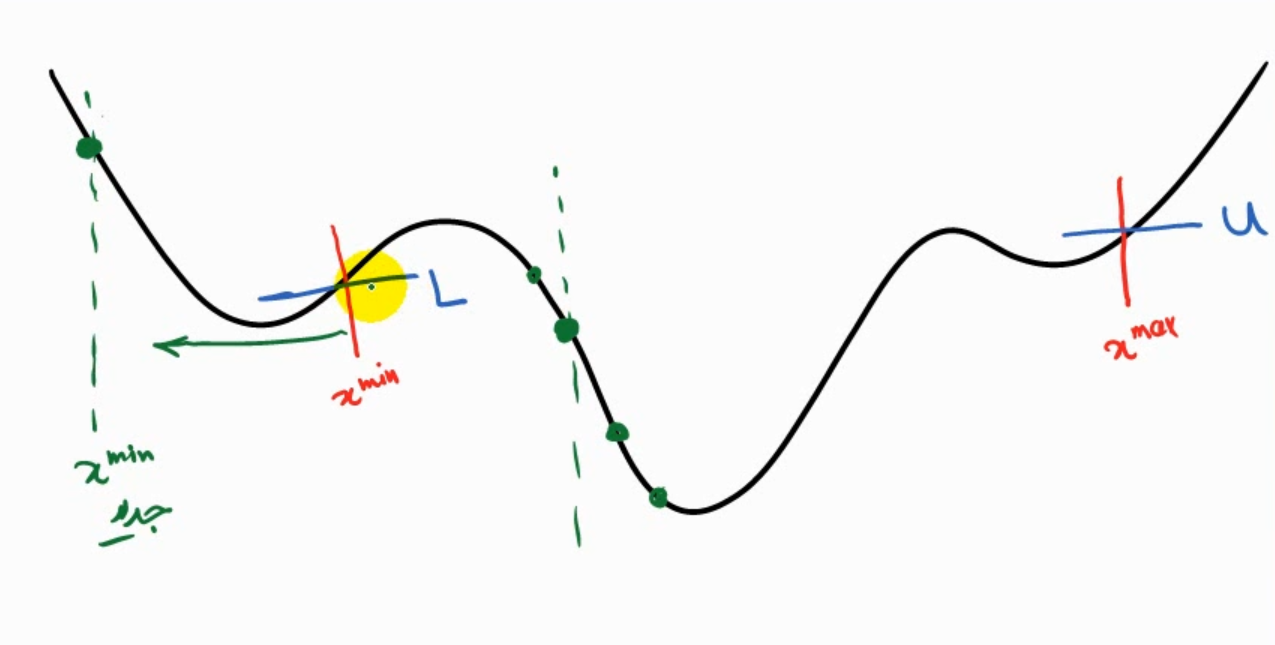
که در اینجا مولفه و همان مقدار حداقل و حداکثری برای متغیر x در حوزه i می باشد.

اگر مقدار حد قبلی پاسخ را از قبل داشته باشیم و مقدار یک پاسخ جدید پذیرفته شده را نیز بپذیریم. می‌توان به مقدار پاسخ سیستم به مولفه j در مومنتوم زمانی بعدی نیز دست پیدا کنیم.

به عبارت دیگر خواهیم داشت:

تنظیم فضای اعتقادی:

ضوابط مربوط به حد بالا و پایین جست و جو و حد بالا و پایین مطلق به شکل زیر محاسبه می گردند:



تصویر 5 شماتیک کلی از نحوه جست و جو وغربال گری در الگوریتم فرهنگی

مولفه‌های تشکیل دهنده فضای اعتقادی عبارت است از :

بهترین پاسخ یافته شده تا کنون

حد بالا و حد پایین بازه بررسی

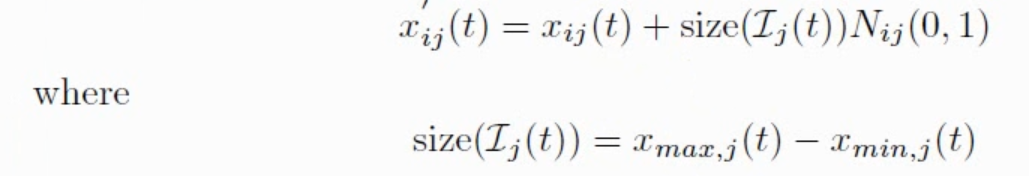
با استفاده از B(t) میتوان یک فضای اعتقادی یا فرهنگ را توصیف کرد.

# تابع تاثیر(Influence Function) :

تابع تاثیر روش‌های تغییر پاسخ ها و یا به عبارتی روش‌های تاثیر فرهنگ را روی جامعه مدل می‌کند. از این روش ها (که اصطلاحا روش‌های تغییر پاسخ‌ها نام دارند) میتوان به موارد زیر اشاره کرد:

## صرفا استفاده از مولفه هنجاری :

می‌خواهیم مولفه j ام پاسخی را تغییر بدهیم. و مقدار جدید آن را با پریم نشان می دهیم. (مقدار قدیمی پریم ندارد)



تصویر 6 استفاده از مولفه هنجاری برای محاسبه تابع تاثیر

جواب جدید برابر است با جواب قدیم به علاوه طول بازه (گام) ضربدر یک عدد تصادفی نرمال استاندارد.

## صرفا استفاده از مولفه وضعی برای تشخیص جهت حرکت:

اگر پاسخ بدست آمده کمتر ازپاسخ قبلی ما باشد، بایستی پاسخ را با مقدار مثبتی جمع کرده تا به مقدار مطلوب برسیم. به همین ترتیب اگر پاسخ بدست آمده بیشتر از پاسخ قبلی ما باشد، بایستی حاصل مقدار جدید از عدد معین مثبتی کم شود تا به نتیجه دلخواه دست پیدا کنیم.

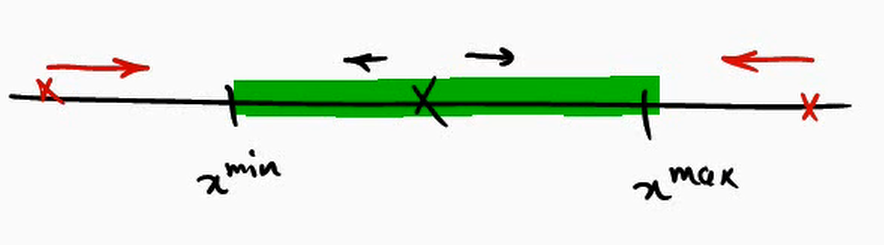
در حالت اول گام مثبت، در حالت دوم گام منفی، و در حالت سوم گام مثبت و منفی با احتمال یکسان میباشد.

## استفاده از مولفه موضعی برای تعیین جهت و مولفه هنجاری برای میزان حرکت:

همان استفاده از مولفه وضعی است، با این تفاوت که برای مقدار دهی سیگما، برخلاف حالت دوم که مقداری ثابت در نظر گرفته می شد، در این روش از یک ضریب و مقدار متغیر برای مقدار‌دهی ضریب تغییر در پاسخ ها استفاده می کنیم.

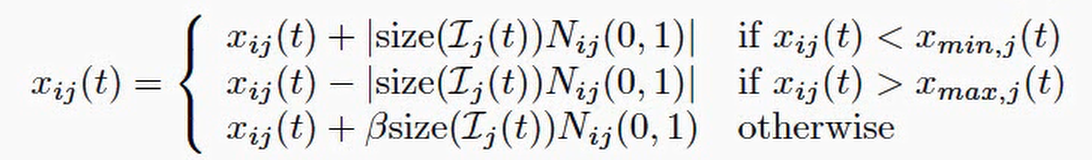
و برای محاسبه پاسخ از همان روابط روش دوم استفاده می‌کنیم و تنها تغییر انجام در نحوه محاسبه مقدار سیگما می‌باشد.

## استفاده از مولفه هنجاری هم برای تعیین جهت و هم برای تعیین میزان گام:

فرض کنید بازه هنجار مساله مورد بررسی ما بازه سبز رنگ نشان داده شده در شکل زیر باشد. اگر ما در خارج از این بازه قرار داشته باشیم بایستی با مقادیر جابه جا کننده مناسب به بازه هنجار مورد بررسی منتقل شویم. بطوریکه که اگر پاسخ مورد بررسی کوچکتر و با کمتر از بازه هنجار باشد بایستی به سمت راست حرکت کرده و به همان ترتیب تصمیم مناسب را برای مقادیر بزرگتر از پاسخ های هنجار اتخاذ کنیم. 

تصویر 7 اسفاده از مولفه هنجاری هم برای تعیین جهت و هم برای تعیین میزان گام

اما زمانیکه در بازه هنجار قرار داریم می‌توانیم به هرجهتی ( چه به سمت پاسخ های بزرگتر و چه به سمت پاسخ های کوچکتر) حرکت کنیم. که این امکان با استفاده از یک ضریب به نام ضریب بتا برای الگوریتم فراهم می‌شود.

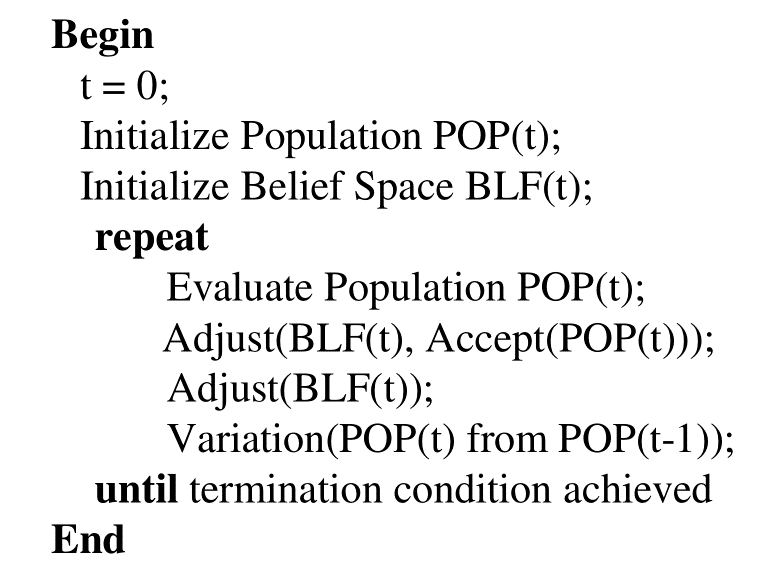


ضریب Ij(t) نیز همان سیگما می باشد.

# مراحل (روند اجرا) الگوریتم فرهنگی:

1. تولید جمعیت اولیه و ارزیابی آن‌ها
2. آماده سازی فضای اعتقادی
3. انتخاب اعضای جمعیت برای تاثیر بر فرهنگ
4. تغییر مولفه های فضای اعتقادی (فرهنگ) توسط اعضای منتخب
5. تولید پاسخ های جدید با تاثیر فرهنگ بر پاسخ های فعلی و ارزیابی آنها
6. بازگشت به مرحله سوم در صورت برآورده نشدن شرایط خاتمه

که این روند بصورت الگوریتمی به شکل زیر میتواند مدل شود:



# پیاده سازی اولیه الگوریتم فرهنگی:

الگوریتم پیاده سازی شده بر اساس استاندارد های بیان شده به پنج قسمت تقسیم بندی می شود که دسته های مختلف آن به شرح زیر است:

1. تعریف مساله
2. تعریف پارامتر‌های الگوریتم
3. آماده سازی (Initialization)
4. حلقه اصلی الگوریتم (پیاده سازی تکرار الگوریتم)
5. نمایش نتایج

## نیازمندی های پیاده سازی:

برای پیاده سازی نیاز به تابع sphere داریم. تابع sphere مجموع مربعات مولفه‌های یک بردار را محاسبه کرده و به دست می دهد.

که مفروضا x یک بردار باشد که حاوی مولفه های زیر باشد، نهایتا تابع f مجموع توان دوم مولفه‌های آن بردار را محاسبه می‌کند.

کد مربوط به تابع sphere به شرح زیر است:

function z=Sphere(x)

z=sum(x.^2);

end

برنامه اصلی را با سه دستور زیر آغاز می کنیم که بیانگر عملیات های زیر می باشد:

* دستور CLC : که صفحه ویرایشگر متن محیط متلب را پاک می کند.
* دستور clear :که حافظه موقت سیستم را از برنامه های پیشین اجرا شده توسط متلب پاک می کند.
* دستور close all : که تمامی توابع، خروجی ها، و نمودار های قبلی ایجاد شده توسط محیط متلب را می بندد.

## قسمت اول – تعریف مساله :

در ابتدا تعریف تابع هزینه یا Cost Function را خواهیم داشت. که تابعی است از متغیر x که خروجی از نوع sphere(x) را به ما خواهد داد. تعداد متغیر های تصمیم را 5 در نظر می‌گیریم. و آن را در یک بردار یک در پنج ذخیره می‌کنیم. البته تفاوت چندانی در انتخاب 1\*5 و یا 5\*1 بودن بردار وجود ندارد، چرا که در این الگوریتم محاسبات ماتریسی وجود ندارد (واضح است که در محاسبات ماتریسی همچون خوشه بندی و تشکیل dataset استفاده از بردار‌ها با ابعاد عمودی بهینه‌تر می باشد). البته برای نمایش‌دادن داده‌ها استفاده از بردار افقی بهتر بنظر می‌رسد. اما در کل در این پیاده سازی نوع بعد بردار تصمیم تاثیری در فرایند محاسبات نخواهد داشت.

%% Problem Definition

CostFunction=@(x) Sphere(x); % Cost Function

nVar=5; % Number of Decision Variables (Unknown variables)

VarSize=[ nVar]; % Decision Variables Matrix Size

VarMin=-10; % Decision Variables Lower Bound

VarMax= 10; % Decision Variables Upper Bound

پس اندازه ماتریسی که حاوی برداری که مولفه‌های آن متغیر های تصمیم الگوریتم خواهد بود 5 عدد اتخاذ گردید. حد پایین و حد بالا بترتیب -10 و +10 انتخاب شدند.

## قسمت دوم – تنظیمات مربوط به الگوریتم فرهنگی:

تعداد دفعات تکرار حلقه تکرار و تعداد اعضای جمعیت از جمله ملزومات تنظیمات اولیه یک الگوریتم تکاملی می باشد که در همین ابتدا مقدار‌دهی می شوند.

پارامتر مهم بعدی نرخ افرادی است که انتخاب می‌شوند تا بر روی فضای اعتقادی تاثیر بگذارند . که این پارامتر با نام نرخ مقبولیت یا acceptance ratio خوانده می‌شود. در نهایت با ضرب کردن این نرخ در تعداد کل جمعیت تعداد جمعیت تاثیر‌گذار بر روی فرهنگ بدست می آید که در الگوریتم با نام nAccept پیاده سازی شده است. و در نهایت مقدار بدست آمده با استفاده از تابع round به یک مقدار صحیح گرد می شود.

%% Cultural Algorithm Settings

MaxIt=500; % Maximum Number of Iterations

nPop=50; % Population Size

pAccept=0.35; % Acceptance Ratio

nAccept=round(pAccept\*nPop); % Number of Accepted Individuals

alpha=0.25;

beta=0.5;

که البته پیاده سازی فوق یک پیاده سازی ایستا (Static) می‌باشد. می توان این نرخ و طبعا این مقدار جمعیت را به شیوه پویا یا Dynamic نیز بدست آورد که در هر حلقه تکرار nAccept بدین صورت بدست آید:

for i=1 : it

nAccept= round(nPop\*gamma/it);

end

## قسمت سوم – آماده سازی :

همانند همه الگوریتم‌های تکاملی هر پاسخ مفروض در فضای مساله دارای دو ویژگی بسیار مهم و حیاتی می باشد:

1. موقعیت (Position) : که موقعیت پاسخ مورد نظر را در فضای جست و جو نشان می دهد. ( یک پاسخی را ارائه می کند.)
2. هزینه (Cost) : هر پاسخ مقداری متناسب با تابع هدف را داراست که با استفاده از این ویژگی در مساله بیان خواهد شد.

در ابتدا یک ساختار خالی برای پذیرش مقادیر پاسخ‌ها تعریف می کنیم. واضحا هر کدام از این پاسخ ها یک موقعیت در فضا دارند.

سپس آماده‌سازی آرایه مربوط به اعضای جمعیت را انجام می‌دهیم. که این کار با استفاده از تابع repmat() که مخفف repeat matrix می باشد انجام می گیرد. لزوم استفاده از این تابع تکرار به اندازه تعداد اعضای جمعیت است. نتیجه نهایی تکرار شدن افقی یا زیر همِ آرایه empty\_individual می باشد. برای مقدار‌دهی این مقادیر از یک حلقه تکرار استفاده می‌کنیم تا به ازای تمامی اعضای جمعیت این فرآیند را انجام دهد.

% Generate Initial Solutions

for i=1:nPop

pop(i).Position=unifrnd(VarMin,VarMax,VarSize);

pop(i).Cost=CostFunction(pop(i).Position);W

end

موقعیت(pop(i).Position) عضو i ام جمعیت یک عدد تصادفی است در بازه تعریف شده جست و جو ، که با استفاده از تابع uniform Random این اعداد تصادفی تولید خواهند شد. و به همان ترتیب یک مقدار هزینه به تناسب با موقعیت آن تولید کرده و ذخیره می‌کنیم.

سپس بلافاصله مقادیر پاسخ ها را مرتب خواهیم کرد. برای مرتب سازی نیاز به مقدار هزینه تمامی اعضای جمعیت داریم. برای اینکار متغیر pop.Cost را بصورت یک ماتریس در میاوریم. که اینکار براحتی با type Casting و اضافه کردن علامت کروشه به عبارت در محیط متلب انجام می پذیرد.

% Sort Population

[~, SortOrder]=sort([pop.Cost]);

pop=pop(SortOrder);

خروجی تابع sort دو مقدار می باشد، مقدار اول که مقدار مرتب شده هزینه‌ها (مقادیر متناظر Cost برای هر پاسخ) می‌باشد. که نیازی به آن در این مقطع نداریم. پس مقدار خروجی اول را Null در نظر می گیریم. که این کار در محیط برنامه نویسی متلب با علامت تیلدا (~) انجام می پذیرد. مقدار خروجی دوم ترتیب قرارگیری اعضای جمعیت می باشد که SortOrder نامگذاری شده است.

pop=pop(SortOrder) نیز اعلام می‌دارد که اگر جمعیت ما همان جمعیت قبلی باشد با این تفاوت که طبق نحوه قرارگیری شان در آرایه مرتب بازچینی شده باشد، جمعیت حاصله یک جمعیت مرتب خواهد بود.

## قسمت چهارم - بروزرسانی بهترین پاسخ یافت شده:

قسمت بعدی بروزرسانی بهترین پاسخ یافت شده می‌باشد:

% Update Best Solution Ever Found

BestSol=Culture.Situational;

یک آرایه نیز به منظور نگهداری بهترین پاسخ ها در هر تکرار در نظر می‌گیریم، در ابتدا مقادیر را با صفر پر می کنیم.

% Array to Hold Best Costs

BestCost=zeros(MaxIt,1);

## قسمت پنجم – آماده سازی فرهنگ:

سپس نوبت به مهم ترین بخش آماده‌سازی ینی آماده سازی فرهنگ می‌رسد. یعنی پیاده سازی و بروزرسانی مولفه های هنجاری. اما در ابتدا لازم است که یک فرهنگ خالی ایجاد شود، سپس به بروزرسانی آن بپردازیم.

شایان ذکر است که در این نوع الگوریتم بخش فرهنگ بسیار مهم تر از پاسخ های فردی موجود در فضای مساله می باشد. به عبارتی پاسخ‌های مساله (در مساله مورد بررسی پاسخ های منفرد (individuals) فقط خودشان را توصیف می‌کنند. این در حالیست که فرهنگ مجموعه‌ای از پاسخ های منفرد (مجموعه‌ای از individual ها) را توصیف می‌کند. پس واضحا یک مدل (در اینجا فرهنگ) می‌تواند بعد‌ها به عنوان مبنایی برای استنتاج قرار گیرد.

قسمت مربوط به آماده‌سازی فرهنگ به شکل زیر می باشد:

% Initialize Culture

Culture.Situational.Cost=inf;

Culture.Normative.Min=inf(VarSize);

Culture.Normative.Max=-inf(VarSize);

Culture.Normative.L=inf(VarSize);

Culture.Normative.U=inf(VarSize);

در ابتدا مقدار هزینه مولفه وضعی مقداری بسیار بزرگ جای‌گذاری می شود. سپس نوبت به جای‌گذاری مقادیر اولیه مولفه هنجاری می رسد.

دقت شود که هر پارامتری که به دنبال کمینه کردن آن می‌باشیم بایستی در ابتدا مقدار خام آن بی نهایتِ مثبت قرار گیرد. و متعاقبا هر پارامتری که به دنبال بیشینه کردن آن هستیم بایستی مقدار اولیه آن بی نهایتِ منفی جای‌گذاری شود.درقسمت بالا تنها پارامتری که بیشترین مقدار آن مد نظر است متغیر Culture.Normative.Max می‌باشد که تنها برای این متغیر مقدار اولیه منفی بی نهایت در نظر گرفته شده است.

## علت وجود اسکریپت AdjustCulture.m در الگوریتم چیست؟

سوالی که پیش می‌آید این است که چگونه می‌توان جمعیت ای منتخب از فضای مساله را به یک فضای فرهنگی برده و مقادیر پاسخ‌های آنان را بروزرسانی کرد؟ در پاسخ به این سوال بایستی تابع‌ای تعریف شود که با ورودی و خروجی فرهنگ، مقادیر خروجی آن یک فرهنگ بوده، مقادیر ورودی آن فرهنگ و جمعیت انتخاب شده (جمعیتی که برای ورود به فرهنگ Accept شده اند) برای ورود به فضای فرهنگی ‌باشد. که این امر در قطعه کد زیر قابل مشاهده می‌باشد:

function Culture=AdjustCulture(Culture,spop)

سپس نوبت به پیاده‌سازی شرط‌های مربوط به پیدا کردن مقادیر بهینه پاسخ‌ها در فضای فرهنگی می‌باشد. که این مهم با استفاده از الگوریتم زیر قابل دستیابی می‌باشد:

for i=1:n

if spop(i).Cost<Culture.Situational.Cost

Culture.Situational=spop(i);

end

for j=1:nVar

if spop(i).Position(j)<Culture.Normative.Min(j) ...

|| spop(i).Cost<Culture.Normative.L(j)

Culture.Normative.Min(j)=spop(i).Position(j);

Culture.Normative.L(j)=spop(i).Cost;

end

if spop(i).Position(j)>Culture.Normative.Max(j) ...

|| spop(i).Cost<Culture.Normative.U(j)

Culture.Normative.Max(j)=spop(i).Position(j);

Culture.Normative.U(j)=spop(i).Cost;

end

end

end

Culture.Normative.Size=Culture.Normative.Max-Culture.Normative.Min;

end

به ازای تمامی عناصر جمعیت (متغیر n) و به ازای تمامی متغیر‌های تصمیم (متغیر nvar) بایستی این تکرار انجام پذیرد. (همه اعضای جمعیت، همه متغیر ها)

برای عضو i ام جمعیت و مولفه j ام آن داریم:

if spop(i).Position(j)<Culture.Normative.Min(j) ...

|| spop(i).Cost < Culture.Normative.L(j)

یا ثابت شود که پاسخ مورد نظر می‌تواند پاسخ بهینه‌تری را نسبت به پاسخ قبلی تولید کند( به نحوی میتوان بهینه تر بود). و یا پاسخ مورد بررسی رکورد‌شکنی کند و به عبارتی از مقدار هنجار پیش بینی شده هم بهینه تر و بهتر باشد. در این حالت خود پاسخ به عنوان پاسخ بهینه بروز رسانی می‌شود و علاوه بر این مقدار هزینه این پاسخ به عنوان هزینه بهینه آن هنجار در آن تکرار ذخیره می شود.

و به همین منوال برای مقادیر بزرگتری و بیشینه نیز کد مربوطه پیاده سازی می شود.

قسمت مربوط به بروزرسانی مولفه وضعی الگوریتم فرهنگی در ماژول AdjustCulture.m نیز به شرح زیر است:

for i=1:n

if spop(i).Cost<Culture.Situational.Cost

Culture.Situational=spop(i);

end

اگر هزینه عضو i ام از جمعیت از مقدار رکورد مولفه وضعی کمتر باشد (رکورد مولفه وضعی را شکسته باشد) در آن صورت عضو i ام جمعیت به مولفه وضعی جدید جمعیت تبدیل می شود.

پس می‌توان بعد از مرتب کردن مقادیر جمعیت در ماژول اصلی CA.mمقدار فرهنگ را با استفاده از ماژول AdjustCulture.m بروزرسانی کرد.

% Adjust Culture

spop=pop(1:nAccept);

Culture=AdjustCulture(Culture,spop);

برای این‌کار در ابتدا آن دسته از جمعیت مورد بررسی مساله را که برای تاثیرگذاری در فرهنگ مقبول واقع شده اند را در متغیر دیگری بنام spop به معنای selected population ذخیره کرده و بروزرسانی فرهنگ جدید بر مبنای اجرای تابع AdjustCulture به ازای ورودی های جمعیت جدید انتخاب شده را انجام می‌دهیم.

قسمت مربوط به تاثیر گذاری فرهنگ بر روی جمعیت و تولید جمعیت جدید و بهبود یافته در حلقه اصلی الگوریتم فرهنگی دنبال می‌شود که به چند شیوه‌ای که قبلا اشاره شد (صرفا استفاده از هنجار، استفاده از مولفه وضعی برای تعیین جهت و مولفه هنجاری برای میزان حرکت،) می‌تواند پیاده‌سازی گردد.

حالت اول- صرفا استفاده از مولفه هنجاری

حالت دوم- صرفا استفاده از متغیر وضعی

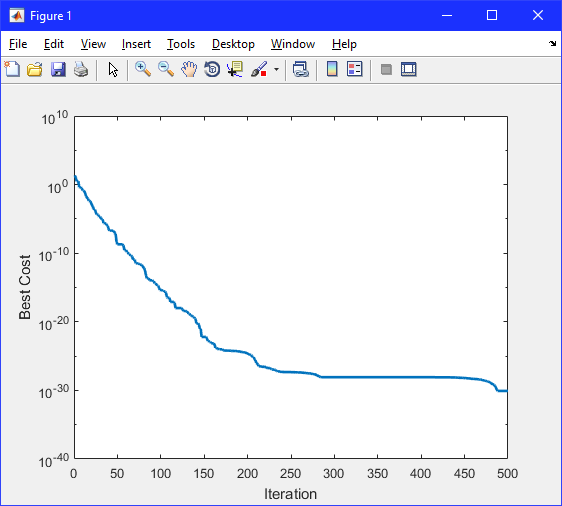
حالت سوم - استفاده از مولفه موضعی برای تعیین جهت و مولفه هنجاری برای میزان حرکت

حالت چهارم - استفاده از مولفه هنجاری هم برای تعیین جهت و هم برای تعیین میزان گام

این پیاده سازی ها در پوشه "پیاده سازی" به انضمام فایل گزارش قابل دسترس می باشند.

## اجرای الگوریتم فرهنگی در بستر نرم افزار متلب

پس از طراحی الگوریتم و پیاده سازی آن در نرم افزار متلب با مقادیر ورودی زیر خروجی مطلوب به شرح زیر می باشد:

* جمعیت: 50
* تعداد تکرار: 500
* نرخ مقبولیت: 0.35
* نرخ آلفا (): 0.25
* نرخ بتا (): 0.5

تصویر 8 خروجی الگوریتم فرهنگی پیاده سازی شده در نرم افزار متلب به ازای پارامتر های ورودی مذکور

# پیاده سازی شبیه ساز تکامل دستور‌های پخت غذا :

## طرح مساله:

از آنجاییکه تمامی الگوریتم های تکاملی شهیر و مطرح در علم کامپیوتر برگرفته از اصول طبیعت و رفتار جمعی حیوانات و یا عناصر طبیعی و یا با استفاده از یک علت و یا معلول اجتماعی پایه‌گذاری می شوند، بر آن شدیم که الگوریتم فرهنگی پیاده‌سازی شده را در قالب یک الگوی اجتماعی/ فرهنگی جدید و قابل درک ارائه کنیم. از این جهت مدلی ارائه شد که روند تکامل فرهنگی را که از ذائقه ها و فرهنگ‌های مختلف نشات میگیرد را در قالب تکامل دستور پخت غذاهای مختلف پیاده سازی می‌کند.

بدین شکل که آشپز‌های مختلف (عامل) هرکدام دستور پخت خاص خود را دارند. به مرور زمان و طی ارتباطات با دیگر آشپز‌ها (با دستور پخت‌ها ، ذائقه‌ها و گرایش غذایی متفاوت) تغییرات (در اینجا به تغییرات در نحوه پخت و یا آماده‌سازی غذاها تکامل گفته می‌شود.) ای در نحوه تهییه دستور غذایی خود خواهند داد تا دستور غذایی واحد که مورد قبول تمامی جوامع آشپزی با تمامی تفاوت هایشان هستند برسیم.

از این نوع پیاده سازی می توان در یافتن علایق مشترک یا نحوه زندگی و یا حتی رئیس جمهور منتخب یک جامعه استفاده نمود.

شبیه ساز تکامل طرز تهییه خوراکی ها از سه قسمت اصلی تشکیل شده است:

1. بخش مربوط به دستور تهییه (Recipe)
2. بخش مربوط به عامل (Agent)
3. بخش مربوط به تولید نسل جدید (Generation)

با اجرای برنامه کاربر قادر به تغییر تمامی پارامتر‌های دخیل در الگوریتم با استفاده از رابط کاربری(GUI) می باشد. و بعد از اجرا نتایج بصورت سلسه مراتبی در فایل متنی ذخیره شده تا اینکه بعد از خاتمه اجرای برنامه نیز نتایج در دسترس باشند. منظور از سلسه مراتبی بودن مراتب ذخیره نتایج این است که اطلاعات مربوط به تمامی عامل‌ها، گروه‌های اجتماعی و هر نسل در هرکدام از شبیه سازی های اجرا شده بصورت فایل متنی ذخیره میشود که بعدها قابل دسترس باشد.

کلاس Recipe.py دستور پخت‌ها را از وبگاه زیر دریافت می کند:

<http://mc6help.tripod.com/RecipeLibrary/RecipeLibrary.htm>

که این دستور پخت‌ها قبلا بارگذاری شده و در دایرکتوری پروژه در پوشه Recipes در دسترس هستند. لازم به ذکر است که دستور پخت‌ها در روند اجرای الگوریتم تاثیر داده نشده اند. دستور پخت‌ها بر اساس ذائقه‌های از پیش تعیین شده‌ی عامل ها دسته بندی می شوند.

پارامتر‌های مهم هر دستور پخت برای ما به شرح زیر است:

1. عنوان (title)
2. مدت زمان پخت یا آماده‌سازی
3. مواد تشکیل دهنده
4. Mutation history یا پیشینه نرخ جهش
5. Evaluation score یا نتیجه ارزیابی

هر عامل در ابتدای کار با داشتن یکی از ذائقه‌های ماهی، گوشت و یا سبزیجات ایجاد می‌شود. در اولین نسل تعداد عامل‌های با ذائقه‌های متفاوت توسط کاربر در محیط کاربری تعیین می‌شود. هر عامل با توجه به ذائقه غذایی خود یکی از دستورات پخت مربوط به ذائقه خود را بصورت تصادفی انتخاب می‌کند. دستور پخت انتخاب شده تمام ویژگی‌های عامل را تعیین می‌کند. بدین شکل که مواد تشکیل دهنده دستور پخت انتخاب شده، مواد تشکیل دهنده محبوب آن عامل خواهند شد. بر اساس مدت زمان لازم برای پخت یا آماده سازی عامل ها ویژگی بنام سرعت تهییه پیدا میکنند که مقادیر "سریع"، "متوسط" و یا "کند" را به خود می گیرد.

بعد از طی کردن انتساب مقادیر و ویژگی‌های هر عامل، عامل غذای انتخاب شده خود را تهییه می‌کند. حتی خواهیم دید که مقادیر مواد اولیه را حذف اضافه و یا کم و زیاد کردن می‌کند. البته روشی که عامل دستور پخت خود را تغییر می‌دهد نیز به خودی خود بصورت تصادفی انتخاب می‌شود که این امر به نرخ MUTATION الگوریتم فرهنگی پیاده‌سازی شده بستگی دارد. هرگونه تغییرات در دستور پخت بوسیله عامل ممکن است، و هر تغییر به عنوان پیشینه تغییرات دستور پخت ذخیره و نگهداری می شود.

بطور دقیق تر فقط یکی از اعمال زیر و آن هم فقط یکبار ( عامل فقط یک بار اجازه دخالت در دستور پخت خود را داراست) بوسیله عامل بر روی دستور پخت خود ممکن است:

1. None کار خاصی انجام نشود
2. اضافه کردن ماده تشکیل دهنده
3. حذف ماده تشکیل دهنده
4. تعویض یا تغییر ماده تشکیل دهنده

در پایان این مرحله، عامل دستور پخت خود را ارزیابی می‌کند. طبیعی است که غذای جدید ویژگی های غذای قبلی را نداشته باشد. و این موضوع در سیر تکامل عامل ها در الگوریتم های تکاملی امری طبیعی محسوب می شود. این عمل در مرحله ای از انجام الگوریتم بنام "مرحله ارتباطات" انجام می‌گیرد. تمامی این مراحل نیز در پیشینه فایل‌های متنی مربوط به هر عامل ذکر خواهد شد.

## مراحل انجام الگوریتم فرهنگی بر روی دستور‌های پخت:

بعد از طی مراحل اولیه، مرحله ارتباطات انجام می‌پذیرد که عامل با ارتباط با دیگر عامل‌ها دستور پخت خود را دچار تغییر می‌کند. تعداد دوستان و هم گروهی‌هایی که عامل می‌تواند با آنها به تبادل اطلاعات بپردازد نیز در رابط کاربری برنامه قابل تغییر می باشد. البته هر عامل می تواند فقط عضو یک گروه اجتماعی باشد. البته عامل ها بطور تصادفی در یکی از گروه‌های اجتماعی (برای بخش ارتباطات) عضو می‌شوند تا در یک گروه عامل‌های مختلفی با ذائقه‌های متفاوت (گوشت قرمز خواران ، ماهی خواران و یا سبزی خوار‌ها) وجود داشته باشد.

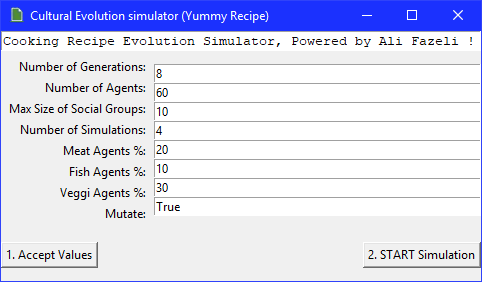
مرحله ارتباطات بگونه ای فرایند انتخاب (selection) را شبیه سازی می‌کند. بطوریکه اعضای فرهنگ (در اینجا عامل‌های آشپزی) بر ازاساس شایستگی‌های خود ارزیابی و انتخاب می‌شوند. این شایستگی بر اساس امتیازی که برای هر عامل در نظر گرفته شده بوجود می آید. در فرایند ارتباطات در گروه‌های اجتماعی (social groups) براساس امتیازی که عامل‌ها به همدیگر اعطا می کنند محبوبیت و شایستگی عامل‌ها در گروه‌های اجتماعی مشخص می‌شود. این شایستگی تحت عنوان دوست داشتن (like) شبیه سازی شده است. بطوریکه یک امتیاز برای هرکدام از مشترکات مواد تشکیل دهنده ، دو امتیاز برای زمان تهییه یکسان و چهار امتیاز برای یکسان بودن دسته بندی خوراکی (گوشت قرمز ، ماهی و یا سبزیجات) در نظر گرفته می‌شود. در هر مرحله از اجرای selection این فرایند امتیازدهی توسط تمامی افراد گروه اجتماعی انجام می‌شود و مجموعه امتیازاتی که اعضای گروه به هریک از هم گروهی های خود می‌دهند امتیاز نهایی هر عامل و شایستگی آن را مشخص می کند. و در نهایت هر عامل با بیشترین میزان امتیاز در هرمرحله، به مرحله بعدی از افراد جمعیت منتقل می شود.

و این فرایند در نسل های بعدی جمعیت نیز تکرار شده و نسل‌های جدید بر اساس شایستگی بیشتر غربال می شوند. طبیعتا دستور پخت عامل غربال شده در هر نسل و هر گروه اجتماعی نشان دهنده ذائقه محبوب و نظر جمعی آن گروه اجتماعی می‌باشد. البته تعداد دستور پخت ها (به تبع آن عامل ها) در طی زمان بشدت کاهش می‌یابد. بدین شکل که به محض غربال یکی از عامل‌ها در یکی از چند گروه اجتماعی ( که بوسیله کاربر تعیین می شوند) شبیه سازی در آن مرحله به پایان می رسد و تنها یک عامل غربال شده از آن گروه به نسل بعدی انتقال می‌یابد و تمامی هم گروهی های آن حذف خواهد شد.

در آخر نتیجه نهایی یک دستور پخت خواهد بود که محبوب ترین و شایسته ترین دستور پخت موجود در تمامی نسل ها می باشد.

## نحوه اجرای الگوریتم:

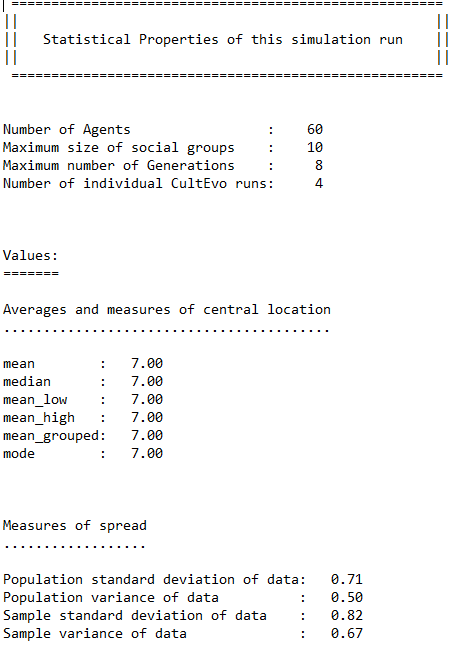
برای اجرای الگوریتم، نیاز به کتابخانه python نسخه 2.7 داریم. با در دست داشتن این کتابخانه، میتوان با اجرای اسکریپت Yummy\_Recipe.py به محیط کاربری شبیه ساز وارد شویم.



تصویر\_ 9 محیط کاربری شبیه ساز

همانطور که در شکل بالا قابل مشاهده می باشد، میتوان شبیه سازی را مقدار دهی و اجرا کرد. در ابتدای کار مقادیر پیشفرض برای مقادیر در فیلد‌های مربوطه از قبل نوشته شده اند. برای تغییر در هرکدام از فیلد ها میتوان مقدار مورد نظر را وارد نمود. در قسمت پایینی محیط امکان تغییر سهم درصدی عامل‌های هر بخش را تعیین کرد، بدین معنی که بطور مثال از مجموع دیتاست دستور پخت lamb.txt و beef.txt چند درصد دستور پخت ها در آزمایش شرکت داشته باشند. و این امر با توجه به ازدیاد دستور پخت ها انجام گردید. و در آخر میتوان تعیین کرد که نرخ جهش در انجام الگوریتم موثر باشد یا خیر.

لازم به ذکر است در صورت تعیین نشدن پارامتر ها و یا خالی گذاشتن آنها مقادیر پیشفرض طبق یک استثنای ورود(I/O Exception) به الگوریتم تزریق می شوند. بعد از وارد کردن مقادیر بر روی دکمه Accept Values کلیک کرده تا تمامی ورودی ها به شکل یک داده ساختار لیست دربیایند. سپس بر روی START Simulation کلیک می‌کنیم.مقادیر انتخاب شده تمامی عامل ها در پوشه Simulation و نتیجه نهایی محاسبات در فایل متنی‌ بنام Stats.txt در مسیر اصلی اسکریپت ها ایجاد خواهند شد.

به عنوان مثال با اجرای الگوریتم با استفاده از مقادیر پیشفرض در خروجی Stats مقادیر زیر را داریم:

تصویر\_ 10 مقادیر خروجی الگوریتم فرهنگی به ازای ورودی های پیشفرض

همانطور که قابل مشاهده است مقادیر ورودی الگوریتم در بخش اول دوباره نشان داده شده و مقادیر میانگین، میانه، مقادیر حداقل ای و حدااکثری داده‌ها نیز نمایش داده شده اند.

در نهایت مقادیر انحراف معیار و واریانس داده های مورد بررسی (در اینجا دستور های پخت) نیز نمایش داده می شوند. اما نکته ای که بیش از هرچیز اهمیت دارد این است که در هر دوره از شبیه سازی در پوشه Simulation در دایرکتوری برنامه اطلاعات مربوط به تمامی عامل‌های درگیر بصورت فایل متنی ذخیره می شوند. در ابتدا به ازای هر عامل یک فایل متنی با نام شماره شناسایی آن عامل ایجاد شده، ذائقه آن ذکر شده و دستور پخت پیشنهادی آن عامل نیز ثبت می شود. سپس در هر شبیه سازی فایل متنی ای بنام Gen\_and\_SocGrps.txt ایجاد شده که اطلاعات جامعی از روند اجرای الگوریتم و تغییرات بوجود آمده بدست می‌دهد. در ابتدا تعداد عوامل موجود به تفکیک بخش مربوطه (گوشت قرمز، ماهی و یا سبزیجات) ذکر می‌شود. سپس تمامی گروه‌های اجتماعی (که در هنگام اجرای الگوریتم تعداد آنها توسط کاربر مقداردهی شده بودند) به همراه شماره شناسایی عامل‌های عضو هر گروه چاپ می‌شود.

بعد از آن به ازای تمامی نسل‌های مختلف، دستور پخت منتخب را بر اساس تعداد امتیاز های هریک(که هرکدام نیز جداگانه چاپ خواهند شد) به خروجی می‌روند.

# نتیجه گیری و کارهای آینده:

از این بررسی میتوان در تحلیل نقاط مشترک یک جامعه دارای ویژگی های متعدد استفاده کرد. این ویژگی در پروژه مطالعه شده دستور پخت غذا های مختلف بوده است. اما یقینا و همانطور که در مقدمه نیز اشاره گردید پتانسیل استفاده در الگوریتم های استخراج علایق و حتی بزرگتر در الگوریتم های تصمیم گیری جمعی نیز مورد استفاده قرار گیرد.

امید است در کارهای آینده به الگوریتم بخش unit testing نیز اضافه شود.

علاوه بر این میتوان الگوریتم را بصورت abstract بازتولید نمود تا بتوان در کارهای آینده با توجه به مساله مورد بررسی الگوریتم را شخصی سازی و به طبع آن بهینه سازی نمود.

# منابع:

1. An Introduction to Cultural Algorithms, Robert G. Reynolds
2. Computational Intelligence, an introduction, Andries P.Engelbrecht. Cultural Algorithm section
3. فیلم آموزشی وبگاه فرادرس، آشنایی با الگوریتم های فرهنگی، به نشانی <http://faradars.org/courses/mvrca9107-cultural-algorithm-in-matlab-video-tutorial>
4. اسلاید اموزشی تشابه الگوریتم ژنتیک و الگوریتم فرهنگی، به نشانی، <https://www.slideshare.net/coddot/cultural-algorithm-genetic-algorithms-related-techniques>