نحوه ی کار کلی الگوریتم:

در اینجا ما یک جامعه ی ۶ تا یی در نظر میگیریم. یعنی در هر نسل ۶ جدول وجود خواهند داشت. که در اولین این جداول به طور رندم مقداردهی میشوند به طوریکه در هز زیرمربع ۹*۹ اعداد از ۱ تا ۹ غیر تکراری وجود داشته باشد(مانند الگوریتم SA که توضیح داده شد) و از آنجایی که نیاز داریم همه ی این جداول را نگه داریم یک آرایه ی ۶*۸۱ نیز تعریف میکنیم که هر سطر آرایه برابر مقادیر یک جدول است و ۶ جدول را خواهیم داشت.

سپس در مرحله بعد باید دو والد از جامعه انتخاب کنیم که برای این کار فیتنس هریک از جدول ها را با فرمول fitness و مقدار هزینه ی هر جدول که مانند الگوریتم SA پیاده سازی شده حساب میکنیم. این فیتنس ها را در بازه هایی در نظر میگیریم سپس یک عدد رندم بین ۰ تا ۱ انتخاب میکنیم و اگر در بازه ی هریک از فیتنس های جدول ها بود ، آن جدول به عنوان والد انتخاب میشود.

پس از انتخاب والدین برای تولید فرزند از خانه ای، رندم کروموزوم های والد شکسته میشوند و باهم جابه جا میشوند و بدین گونه فرزند جدید به وجود میآید. در مرحله بعد فرزند باید دچار جهش شود. در این الگوریتم جهش ۲۰درصد است. یعنی ۲۰ درصد ژن های هر کروموزوم جهش پیدا میکنند. پس۱۶ عدد در یک رشته باید جابجا شوند.

پس از آن جامعه ی جدید بعدی تولید میشوند و میشود الگوریتم را باز برای جامعه ی جدید تکرار کرد.

شرط اتمام الگوریتم رسیدن الگوریتم به یک minFitness ای هست که ابتدا برایش تعریف میکنیم. یعنی اگر فیتنس جداول به این مقدار رسید یعنی به جامعه ی هدفمان رسیده ایم و دیگر الگوریتم را ادامه نمیدهد.

جزئیات الگوریتم:

این الگوریتم از ۷ تابع تشکیل شده:

تابع cost دقیقن مانند الگوریتم SA است. به ازای هر سطر و ستون و زیرمربع جدولی که به عنوان ورودی میبیند اگر عدد تکراری در هریک از آنها پیدا کرد متغیر sscore را یکی بالا میبرد و در آخر این متغیر را به عنوان مجموع error هایی که جدولمان دارد یا هزینه ای که دارد برمیگرداند.

تابع initialize نیز دقیقن مانند SA مقدار دهی اولیه میکند بدین صورت که آرایه ی خالی را و یا جدول خالی را با اعداد تصادفی پر میکند. بدین صورت کار میکند که به ازای هر سطر و ستون و زیرمربع جدول حلقه داریم و داخل هر یک از خانه ها برای آن خانه یک عدد رندم از ۰ تا ۹ انتخاب میکند و چک میکند که برابر ۰ نباشد و سپس چک میکند که آن عدد انتخابی در آن زیرمربع تکراری نباشد. اگر در زیرمربع تکراری بود یک عدد دیگر انتخاب کرده و از اول چک میکند. اگر هم نبود به عنوان مقدار آن خانه قرار داده میشود.

تنها تفاوتی که هست این است که در آخر تابع cos در GA باید آن جدولی که مقداردهی کرده ایم را داخل آرایه ی ۶ در ۸۱ ای مان بریزیم تا ذخیره شود. یعنی هربار مقداردهی را به آرایه ی ۹ در ۹ میکند و با آرایه ی ۶ در ۸۱ هر جدول را ذخیره میکند که بعدن به آن دسترسی داشته باشیم. و خانه ی آخر هر کروموزوم ۸۱ تایی را نیز برابر با cost آن جدول میگذاریم.

تابع print هم که به ترتیب از اول آرایه ی دوبعدی شروع به چاپ جدول میکند و سه تا سه تا خط میکذارد که زیرمربع ها مشخص شوند.

تابع chooseParent در ابتدا برای هر جدول fitness را با در دست داشتن cost و فرمول 1/1+cost حساب میکنیم و و در آرایه ای میگذاریم. اینجا فیتنس هایی که محاسبه کرده را چک میکند که اگر به minFitness رسیده بودیم continueAlg رو false کرده و دور بعد دیگر الگوریتم را ادامه نمیدهد.

در گام بعدی عدد رندمی انتخاب میکند و با توجه به فیتنس های جداول این عدد در رنج فیتنس هرکدام از جداول بود آن جدول به عنوان والد اول انتخاب میشود، برای انتخاب والد دوم قبل از انتخاب چک میکند که والد انتخابی همان والد اولی نبوده باشد. اگر بود عدد رندم دیگری انتخاب و دوباره تلاش میکند.

در آخر این تابع crossover صدا زده میشود.

تابع crossover دو والد انتخابی توسط تابع قبلی را به عنوان ورودی میگیرد و یک نقطه ی شکست به طور اتفاقی از ۱ تا ۸۱(طول کروموزوم) انتخاب میکند و از آن نقطه ژن های والدین را جابجا میکند تا فرزندان جدید نسل بعد تشکیل شوند.

پس از اتمام تشکیل فرزندان آنها را جهش میدهد و ChroCostupdate را صدا میزند.

تابع جهش یا mutation با نرخ ۲۰ درصد، ۱۶ ژن (که به طور اتفاقی انتخاب میکند) از هر کروموزوم فرزند را با یک عدد تصادفی از ۱ تا ۹ جایگزین میکند.

در آخر تابع crossCostupdate برای این هدف است که cost های هر کروموزوم که در خانه ی آخر آرایه ی آن ذخیره میکردیم ، برای جامعه ی جدید و فرزندان تشکیل شده آپدیت کند.

برای این کار ابتدا کروموزوم را (یعنی هر جدول را) به آرایه ی ۹ در ۹ و فرم جدول میبرد ت باتواند به عنوان ورودی به تابع cost بدهد و سپس خانه ی آخر کروموزوم ها را به روز رسانی میکند و اگر به نسل جواب رسیده باشیم آنرا چاپ میکند.

در main ما به تعداد اندازه ی جامعه (که اینجا ۶ است) مقدار دهی اولیه میکنیم تا آرایه ی کروموزوم هایمان پر شود و چاپ میکنیم هر جدول را و سپس تا وقتی که به مقدار fitness مورد نظر نرسیدیم برای جامعه ی بعدی والد انتخاب میکنیم.