### 5. *PSO-GA* хибридни алгоритам

### У овом поглављу предложен је хибридни алгоритам [3] добијен коришћењем *particle swarm* оптимизације и генетског алгоритма, који је даље коришћен за решавање оптимизационих проблема са ограничењима, а његова решења су анализирана и упоређена са решењима других аутора, добијених њиховим верзијама еволутивних алгоритама.

### Генетски алгоритам

### Генетски алгоритам [5] је еволутивни алгоритам претраге заснован на хеуристици и природној селекцији. Први пут је предложен 1960. од стране Тома Холанда, и до сада је широко испитиван и коришћен у разним инжењерским дисциплинама. Фундаментални концепт алгоритма је базиран на тези „опстанка најприлагођенијих“ Чарлса Дарвина. У алгоритму, претпостављени скуп решења (*популација*)*,* пролази кроз селекцију која омогућава варијабилност, а користи технике инспирисане природном селекцијом, као што су мутација и рекомбинација (*кросовер*). Свако појединачно решење(*индивидуа*) је оцењено његовом вредношћу у функцији претраге (*фитнес*), од ког зависи да ли ће решење учесвовати у креирању нове итерације популације (*генерације*). Псеудо код је приказан у алгоритму 1.

### Опис хибридног алгоритма

Мотивација иза креирања *PSO-GA* хибрида је свакако идеја да се споје предности генетског алгоритма и *particle swarm* оптимизације. Оба алгоритма имају своје предности и мане. Угенетичком алгоритму, уколико индивидуа није селектована, информације које је она носила се губе заувек, што значи да постоје веће шансе да се заглави у локалном оптимуму, односно слабије је робусности. *PSO* не уништава честице које се покажу лоше у тражењу решења, што га чини робуснијим, мада оне расипају доста ресураса, што чини конвергенцију спором. Дакле, основна идеја комбинације ова два алгоритма јесте спајање могућности друштвеног мишљења у *PSO*, са предностима локалне претраге у *GA.* Пошто су и један и други базирани на популацији, додатно се олакшава њихово комбиновање.

Алгоритам почиње из фазе иницијализације, у којој се иницијализују честице и њихове брзине насумично преко простора претраге, односно свака честица *xi* насумично узима позицију из униформне расподеле , у рангу , где ипредстављају доње и горње ограничење. Вектор брзине (линк) се састоји из два фактора, личног и друштвеног, односно заснован је на знању сваке честице, односно најбољем положају у којем је честица била (*personal best - pbest*), као и на целокупном знању читавог јата, односно најбољој позицији у којем се јато налазило (*global best - gbest*). У таквој конфигурацији свака честица узима у обзир своје лично искуство, као и искуства њених суседа. Након рачунања брзине, свака честица мења своју позицију пратећи једначину (линк).

Када се оправи нова генерација честица, са унапређеним положајима сваке честице, одређени број честица се селектује и над сваком честицом се примењује *GA* засебно. Број честица који се селектује је одређен формулом (линк). Након што се из *GA* популације изабере најбоља честица, *GA* има задатак да направи нову популацију смењујући тачке у тренутној популацији бољим тачкама користећи генетске принципе, и то примењујући операторе селекције, мутације и рекомбинације. Селекција је примењена методом точка рулета (*roulette wheel selection*), а рекомбинација једном тачком раздвајања (*one point crossover*). Након селекције, мутације и рекомбинације, примењена је и форма елитизма, за очување најбољих решења у популацији, дефинисана формулом (линк).

Након евалуације нове популације, величина популације и максимални број итерација се ажурира узимајући у обзир тренутну итерацију *PSO* алгоритма (линк).

Кроз понављање процеса репродукције популације, популација се води ка глобалном оптимуму. Репрезентација алгоритма је приказана на слици (линк).

### Оптимизациони проблеми над којим се тестира алгоритам

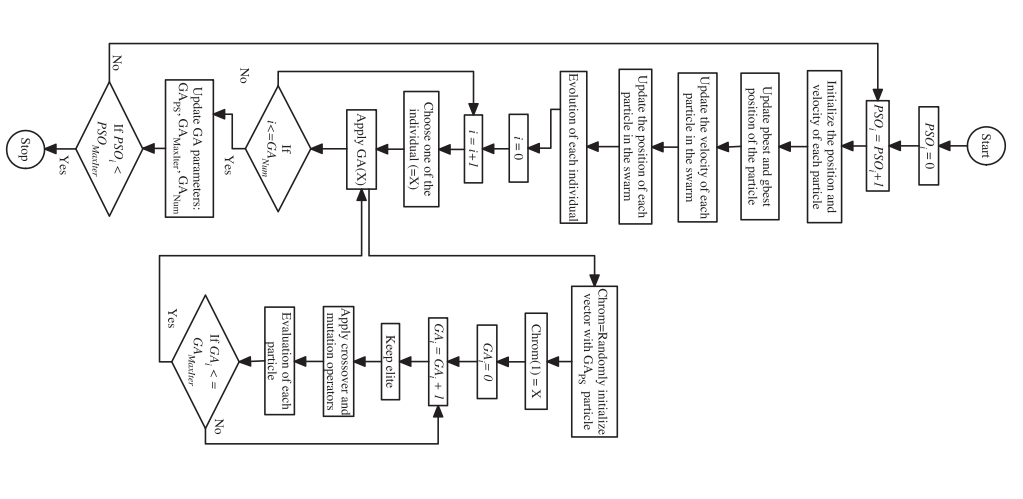
Представљени оптимизациони проблеми јесу нелинеарни оптимизациони проблеми са ограничењима. Главни задатак у решавању ових проблема јесте баратање са ограничењима. Најуобичајенији начин за баратање ограничења у заједници која се бави еволутивним алгоритмима је увођење пенала. У том случају, уколико пронађено решење не задовољава ограничења, решење се „казни“ тако да не учествује у претрази. Упркос великој популарности, увођење пенала има многе недостатке, од којих је главни то што има превише параметара, и тражење одговарајуће комбинације ограничења није лак посао, а такође и узима превише времена. Решење је коришћење пеналних функција без параметара (линк).

#### Химелблауов нелинеарни оптимизациони проблем

Овај проблем је оригинално предложио Химелблау (линк), и он је широко коришћен за упоређивање ефикасности различитих еволутивних алгоритама. Проблем је дефинисан као петодимензионални, са шест нелинеарних ограничеања типа неједнакости и десет граничних услова. Дефинисан је на следећи начин (линк). У табели (линк) се може видети како решење *PSO-GA* хибрида стоји у поређењу са решењима пронађеним другим алгоритмима. У табелама су приказана решења разних метода као што су: *GA* (линк), хармонијска претрага (*harmony search*) (линк), *PSO* (линк), кукавичја претрага (*cuckoo search*) (линк), симплекс (*simplex search*) (линк) и *PSOa*, *PSOstr* (линк). Такође се да приметити да решење (линк) не задовољава ограничење , тако да није валидно. Примећује се да препоручена метода проналази најбоље решење , а вредност функције .

На основу изнетих података да се закључити да је *PSO-GA* препоручена метода по природи врло робусна и има најбољи квалитет претраге. Штавише, најгори резултат добијен препорученом методом је и даље бољи од било ког решења добијених осталим методама претраге.

Алгоритам 1 – псеудо код генетичког алгоритма



Слика 1 - *PSO-GA* алгоритам

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метода | Променљиве | | | | | Решење | Ограничења | | |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Табела 1