

**Nekonvencionalni računalni postupci**

Izvještaj za prvu laboratorijsku vježbu:

Rješavanje Knapsack problema pomoću genetskih algoritama

Autor: Grgo Lovrić

Kako bismo riješili ovaj problem prvo ćemo definirati komponente potrebne za provođenje algoritma:

1. Gen – jedan gen predstavlja jedan predmet u ruksaku i svaki predmet opisujemo sa dva svojstva: težinom i vrijednosti. Ova svojstva poprimaju integer vrijednosti iz određenog intervala.
2. Kromosom – jedan kromosom predstavlja jedan ruksak tj. jednu kombinaciju predmeta u ruksaku. Svaki kromosom je vektor oblika (1, N) gdje je N sveukupni broj predmeta, a svaki element vektora predstavlja količinu određenog predmeta koja poprima vrijednost (također integer) od 0 do 3.
3. Populacija – skup jedinki tj. skup kromosoma. Definiramo ju kao matricu čiji su redci kromosomi što znači da je ona dimenzije (M, N) gdje je M broj jedinki u populaciji.
4. Fitness funkcija – funkcija kojom ćemo mjeriti koliko je jedinka ''dobro'' rješenje. Kao parametre prima gene (predmete), populaciju (ruksake) te maksimalnu težinu ruksaka. Definiramo ju kao matrično množenje matrice populacije i vektora gena. U slučaju da sveukupna suma težina predmeta iz jednog ruksaka ne prelazi maksimalnu težinu ruksaka zapisujemo sumu vrijednosti predmeta kao rezultat, u suprotnome rezultat je 0. Ne postoji gornja granica koju suma vrijednosti predmeta može dosegnuti, već tražimo što je moguće veću vrijednost.

Postavke za sve konfiguracije

Max. težina predmeta – 10

Max. vrijednost predmeta – 100

Operator selekcije – K-turnirska selekcija za k = 2

Algoritam završava sa izvedbom kada se u setu rješenja nalazi samo jedna vrijednost, tj. kada su za cijelu populaciju rezultati fitness funkcije isti.

Konfiguracija 1

Maksimalni broj evolucija - 31

Broj predmeta (N) – 10

Broj ruksaka (M) – 10

Težina ruksaka – 50

Operator rekombinacije – Single\_Point\_Crossover

Postotak rekombinacije – 90%

Operator mutacije – Bit\_Flip\_Mutation

Postotak mutacije – 10%

Elitizam – NE

|  |  |  |  |
| --- | --- | --- | --- |
|  | Rješenje | Vrijeme izvršavanja | Iznos fitness funkcije |
| 1. | DA | 0,046 s | 643 |
| 2. | DA | 0,041 s | 595 |
| 3. | DA | 0,069 s | 469 |
| 4. | DA | 0,052 s | 689 |
| 5. | DA | 0,118 s | 716 |

Prosječne vrijednosti:

Postotak dolaska do rješenja: 100%

Prosječno vrijeme izvršavanja: 0,065 s

Prosječan iznos fitness funkcije: 622.4

Konfiguracija 2

Maksimalni broj evolucija - 139

Broj predmeta (N) – 10

Broj ruksaka (M) – 10

Težina ruksaka – 50

Operator rekombinacije – Single\_Point\_Crossover

Postotak rekombinacije – 70%

Operator mutacije – Bit\_Flip\_Mutation

Postotak mutacije – 30%

Elitizam – NE

|  |  |  |  |
| --- | --- | --- | --- |
|  | Rješenje | Vrijeme izvršavanja | Iznos fitness funkcije |
| 1. | DA | 0,336 s | 579 |
| 2. | DA | 0,376 s | 509 |
| 3. | DA | 0,139 s | 989 |
| 4. | DA | 0,119 s | 656 |
| 5. | DA | 0,507 s | 601 |

Prosječne vrijednosti:

Postotak dolaska do rješenja: 100%

Prosječno vrijeme izvršavanja: 0,295 s

Prosječan iznos fitness funkcije: 666.8

Konfiguracija 3

Maksimalni broj evolucija - 16

Broj predmeta (N) – 10

Broj ruksaka (M) – 10

Težina ruksaka – 50

Operator rekombinacije – Two\_Point\_Crossover

Postotak rekombinacije – 90%

Operator mutacije – Inverse\_Mutation

Postotak mutacije – 10%

Elitizam – NE

|  |  |  |  |
| --- | --- | --- | --- |
|  | Rješenje | Vrijeme izvršavanja | Iznos fitness funkcije |
| 1. | DA | 0,033 s | 754 |
| 2. | DA | 0,032 s | 206 |
| 3. | DA | 0,095 s | 828 |
| 4. | DA | 0,061 s | 536 |
| 5. | DA | 0,052 s | 862 |

Prosječne vrijednosti:

Postotak dolaska do rješenja: 100%

Prosječno vrijeme izvršavanja: 0,055 s

Prosječan iznos fitness funkcije: 637.2

Konfiguracija 4

Maksimalni broj evolucija - 140

Broj predmeta (N) – 10

Broj ruksaka (M) – 10

Težina ruksaka – 50

Operator rekombinacije – Two\_Point\_Crossover

Postotak rekombinacije – 70%

Operator mutacije – Inverse\_Mutation

Postotak mutacije – 30%

Elitizam – NE

|  |  |  |  |
| --- | --- | --- | --- |
|  | Rješenje | Vrijeme izvršavanja | Iznos fitness funkcije |
| 1. | DA | 0,121 s | 812 |
| 2. | DA | 0,160 s | 864 |
| 3. | DA | 0,331 s | 567 |
| 4. | DA | 0,483 s | 620 |
| 5. | DA | 0,189 s | 456 |

Prosječne vrijednosti:

Postotak dolaska do rješenja: 100%

Prosječno vrijeme izvršavanja: 0,257 s

Prosječan iznos fitness funkcije: 663.8

Konfiguracija 5

Maksimalni broj evolucija - 11

Broj predmeta (N) – 10

Broj ruksaka (M) – 10

Težina ruksaka – 50

Operator rekombinacije – Single\_Point\_Crossover

Postotak rekombinacije – 90%

Operator mutacije – Bit\_Flip\_Mutation

Postotak mutacije – 10%

Elitizam – DA

|  |  |  |  |
| --- | --- | --- | --- |
|  | Rješenje | Vrijeme izvršavanja | Iznos fitness funkcije |
| 1. | DA | 0,021 s | 671 |
| 2. | DA | 0,034 s | 729 |
| 3. | DA | 0,039 s | 423 |
| 4. | DA | 0,016 s | 443 |
| 5. | DA | 0,016 s | 865 |

Prosječne vrijednosti:

Postotak dolaska do rješenja: 100%

Prosječno vrijeme izvršavanja: 0,025 s

Prosječan iznos fitness funkcije: 626.2

Konfiguracija 6

Maksimalni broj evolucija - 51

Broj predmeta (N) – 10

Broj ruksaka (M) – 10

Težina ruksaka – 50

Operator rekombinacije – Single\_Point\_Crossover

Postotak rekombinacije – 70%

Operator mutacije – Bit\_Flip\_Mutation

Postotak mutacije – 30%

Elitizam – DA

|  |  |  |  |
| --- | --- | --- | --- |
|  | Rješenje | Vrijeme izvršavanja | Iznos fitness funkcije |
| 1. | DA | 0,098 s | 544 |
| 2. | DA | 0,185 s | 1054 |
| 3. | DA | 0,131 s | 472 |
| 4. | DA | 0,068 s | 249 |
| 5. | DA | 0,054 s | 155 |

Prosječne vrijednosti:

Postotak dolaska do rješenja: 100%

Prosječno vrijeme izvršavanja: 0,107 s

Prosječan iznos fitness funkcije: 494.8

Konfiguracija 7

Maksimalni broj evolucija - 8

Broj predmeta (N) – 10

Broj ruksaka (M) – 10

Težina ruksaka – 50

Operator rekombinacije – Two\_Point\_Crossover

Postotak rekombinacije – 90%

Operator mutacije – Inverse\_Mutation

Postotak mutacije – 10%

Elitizam – DA

|  |  |  |  |
| --- | --- | --- | --- |
|  | Rješenje | Vrijeme izvršavanja | Iznos fitness funkcije |
| 1. | DA | 0,028 s | 443 |
| 2. | DA | 0,031 s | 543 |
| 3. | DA | 0,008 s | 394 |
| 4. | DA | 0,015 s | 265 |
| 5. | DA | 0,015 s | 673 |

Prosječne vrijednosti:

Postotak dolaska do rješenja: 100%

Prosječno vrijeme izvršavanja: 0,019 s

Prosječan iznos fitness funkcije: 463.6

Konfiguracija 8

Maksimalni broj evolucija - 59

Broj predmeta (N) – 10

Broj ruksaka (M) – 10

Težina ruksaka – 50

Operator rekombinacije – Two\_Point\_Crossover

Postotak rekombinacije – 70%

Operator mutacije – Inverse\_Mutation

Postotak mutacije – 30%

Elitizam – DA

|  |  |  |  |
| --- | --- | --- | --- |
|  | Rješenje | Vrijeme izvršavanja | Iznos fitness funkcije |
| 1. | DA | 0,108 s | 662 |
| 2. | DA | 0,033 s | 593 |
| 3. | DA | 0,113 s | 561 |
| 4. | DA | 0,216 s | 606 |
| 5. | DA | 0,091 s | 437 |

Prosječne vrijednosti:

Postotak dolaska do rješenja: 100%

Prosječno vrijeme izvršavanja: 0,112 s

Prosječan iznos fitness funkcije: 571.8

Iz priloženih mjerenja vidimo da konfiguracije 2 i 4 daju najbolje rezultate. Kako bismo odlučili koju konfiguraciju ćemo koristiti za usporedbu sa konvencionalnim algoritmom povećati ćemo brojeve N i M na 20 te analizirati rezultate (3 mjerenja za svaku konfiguraciju).

Konfiguracija 2

Maksimalni broj evolucija - 736

Broj predmeta (N) – 20

Broj ruksaka (M) – 20

Težina ruksaka – 150

Prosječno vrijeme izvršavanja – 4.744s

Prosječna vrijednost fitness funkcije – 1494.5

Konfiguracija 4

Maksimalni broj evolucija - 4404

Broj predmeta (N) – 20

Broj ruksaka (M) – 20

Težina ruksaka – 150

Prosječno vrijeme izvršavanja – 11.065s

Prosječna vrijednost fitness funkcije – 1995.6

Radi kraćeg izvršavanja odabiremo konfiguraciju 2 za usporedbu. Za različite vrijednosti N ćemo usporediti konvencionalni algoritam te naš genetski algoritam s konfiguracijom 2 i konstatnim brojem ruksaka M = 10.

Konvencionalni algoritam

|  |  |
| --- | --- |
| **N** | Vrijeme izvršavanja |
| 10 | 0,065s |
| 100 | 0,077s |
| 1000 | 2,053s |
| 10000 | 211,121s |
| 100000 | ? |

**Genetski algoritam**

|  |  |
| --- | --- |
| **N** | Vrijeme izvršavanja |
| 10 | 0,213s |
| 100 | 0,215s |
| 1000 | 0,373s |
| 10000 | 1,752s |
| 100000 | 43,241s |

**Linijski prikaz konvencionalni vs. genetski**

Iz priloženog vidimo da već oko N = 1000 genetski algoritam nadmašuje konvencionalni.

Zaključak

U rješavanju problema kao što su Knapsack problem ili problem trgovačkog putnika za velike brojeve čvorova ili predmeta u ruksaku konvencionalni algoritmi pretraživanja postaju vrlo neefikasni. Genetski algoritmi pružaju nam mogućnost da ove probleme rješimo nešto manje efikasnije, sa relativno dobrim rješenjima i u razumnom vremenu.

Ideja za implementaciju genetskog algoritma:

<https://machinelearningmastery.com/simple-genetic-algorithm-from-scratch-in-python/>

Konvencionalni algoritam:

<https://www.analyticsvidhya.com/blog/2022/05/knapsack-problem-in-python/>