

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ
ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО
АНАЛІЗУ

Кафедра математичних методів системного аналізу

ЗВІТ

про виконання лабораторної роботи № 2

з дисципліни «Еволюційні методи оптимізації»

з теми «Дослідження генетичного алгоритму
оптимізації багатоекстремальних функцій у дійсному просторі»

Виконала:

Студентка 3 курсу ІПСА

групи КА-24

Спекторовська Лада

Варіант №36

Мета роботи:

Побудувати процедуру розв'язання багатокритеріальної задачі генетичним алгоритмом і дослідити ефективність методу при зміні його параметрів або елементів.

Програма реалізована власноруч мовою програмування Python. При виконанні було використано бібліотеки: NumPy, Pandas, Matplotlib, Random, Copy.

Використані параметри та елементи методу:

Параметри, що досліджуються: спосіб відбору батьківських точок для кросоверу або мутації (R), коефіцієнти відбору/кросоверу/мутації (C)

- Розмір популяції: 200.
- Коефіцієнти відбору/кросоверу/мутації: 20% - відбір, 70% - кросовер, 10% - мутація(№1); 10% - відбір, 80% - кросовер, 10% - мутація(№2); 30% - відбір, 50% - кросовер, 20% - мутація(№3); 10% - відбір, 30% - кросовер, 60% - мутація(№4).
- Спосіб генерації початкової популяції: рівномірний випадковий розподіл.
- Спосіб відбору точок для переходу в наступне покоління: ранговий (за Парето-фітнесом), 20% еліти (20 індивідів).
- Спосіб відбору батьківських точок для кросоверу або мутації: випадковий з еліти.
- Спосіб кросоверу: природний рівномірний.
- Спосіб мутації: гаусівська мутація.
- Максимальна кількість поколінь: 40.
- Критерій завершення і його параметри: максимальна кількість поколінь – 40.

Програмна реалізація:

Було взято за основу програмну реалізацію минулої лабораторної роботи.

Ключові етапи реалізації:

1. Генерація початкової популяції. Створюється 200 випадкових пар (x, y) у межах області пошуку $[-10, 10]$.
2. Обчислення значень функцій та оцінка Парето-домінування. Для кожної особини і обчислюється $f_1(x_i, y_i), f_2(x_i, y_i)$. Потім визначається, скільки інших особин її домінують (Парето-фітнес). Чим менше значення $\text{fitness}[i]$, тим ближче особина до Парето-фронту.
3. Відбір еліти (елітний відбір). Обираються $\text{selection_rate} * \text{pop_size}$ особин з найкращим Парето-фітнесом (тобто ті, які або не домінуються, або домінуються дуже мало).

4. Генерація нового покоління. Кросовер, мутація, контроль області пошуку: `pr.clip(...)` гарантує, що координати залишаються в межах $[-10,10][-10, 10][-10,10]$
5. Завершення після `gen` поколінь. Алгоритм завершується після 40 поколінь (`gen = 40`). Оцінюється фінальна популяція, і вибираються ті, які мають `fitness[i] == 0` — тобто не домінуються жодною іншою.

Результати тестових запусків:

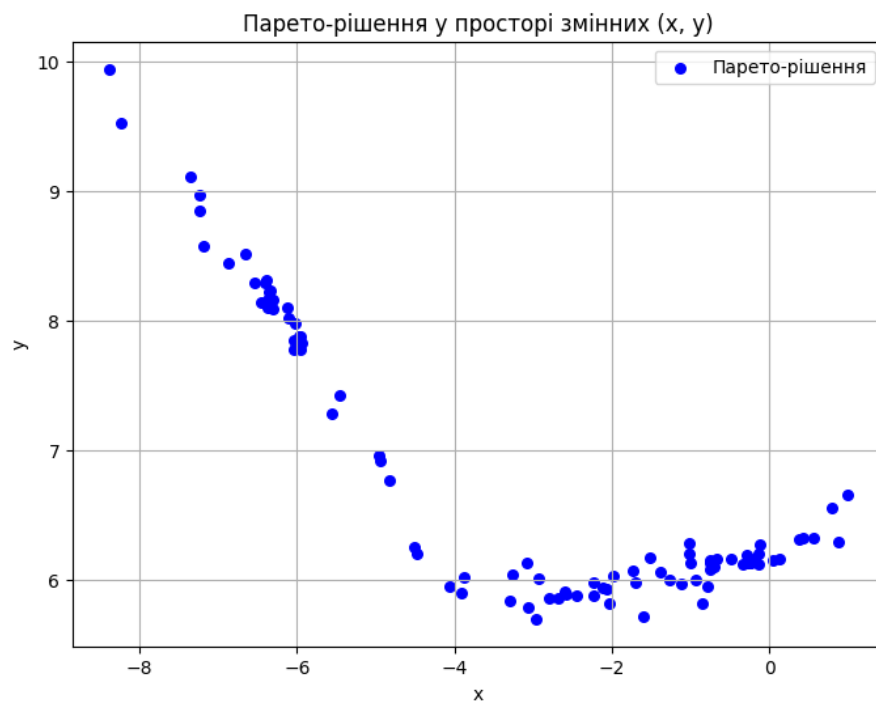
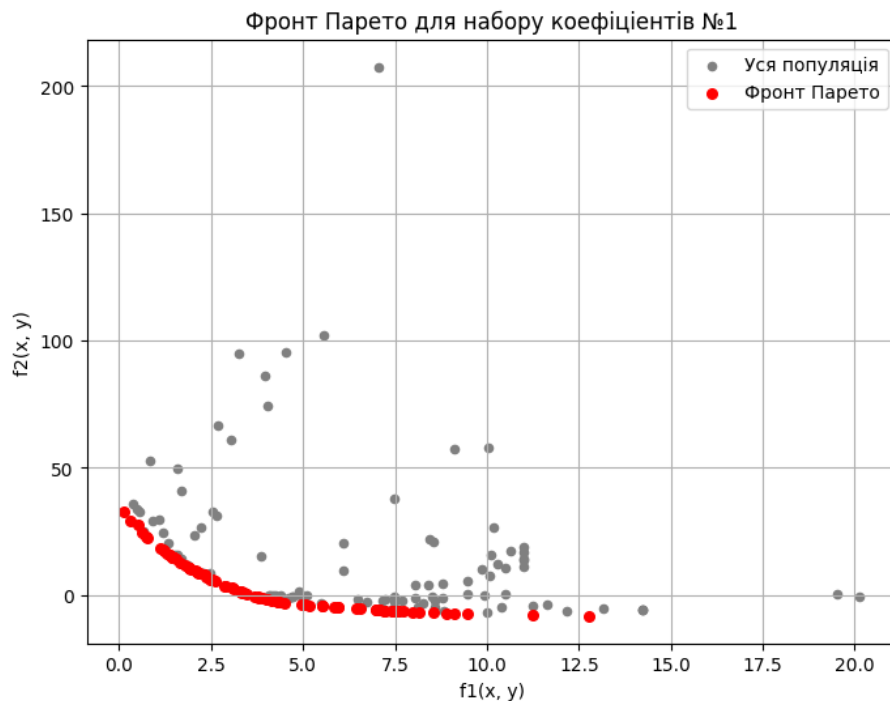
Коефіцієнти відбору/кросоверу/мутації: 20% - відбір, 70% - кросовер, 10% - мутація(№1).

x	y	f1	f2
-6,408887437	8,293248436	7,59531704	-6,395515059
-6,875062809	8,442970922	8,145675275	-6,68835941
-6,019086368	7,980349335	7,084578849	-6,01758581
-2,108194857	5,934390904	3,309121813	1,226797146
-4,94537375	6,915988833	5,94537794	-4,944510277
0,796379834	6,55292117	0,285329597	29,41140106
-0,274396083	6,129021063	1,483093346	14,58373766
-6,374526114	8,161141792	7,496760245	-6,328993245
-3,871455089	6,021827809	4,964532284	-3,848843134
-3,30015089	5,833790744	4,51010171	-3,015379396
-4,831417485	6,761404091	5,831695353	-4,82651561
-0,283766109	6,188721821	1,44265248	14,96491301
-3,265123185	6,037528306	4,364562219	-2,668513513
0,126629471	6,160536905	1,122219122	18,50642501
-0,240664885	6,130070059	1,453259252	14,88680772
-7,236295586	8,842417116	8,908378127	-7,081155337
-2,068048664	5,930940154	3,274006379	1,402316038
-2,22511279	5,981899267	3,387586912	0,861185936
-8,382489742	9,940527615	12,75917633	-8,18715922
-6,313141116	8,161746993	7,436639329	-6,290220936
-0,704437501	6,100161543	1,886991325	10,82650427
-5,933805071	7,828261168	6,967658967	-5,922665555
-4,960146408	6,955429147	5,960146739	-4,960124155
-0,743876267	6,153460752	1,885380897	10,88139009
-1,376137	6,055524312	2,538060732	5,802979368
-8,235101355	9,527814167	11,23019186	-7,73484619
-0,251273546	6,128135603	1,464073	14,77878586
0,383497459	6,3096669	0,779220518	22,40928916
-6,650287247	8,517616151	7,989457762	-6,632685627
0,42636464	6,326328393	0,731452194	23,0144557
-2,232336745	5,881140092	3,466283743	0,48621573
-2,797593721	5,861025581	4,013054778	-1,666706399
-5,962690891	7,877236996	7,005088405	-5,955388523

-6,343767189	8,231974212	7,498967484	-6,331269519
-6,365419183	8,097213059	7,463157399	-6,293484658
-2,582848655	5,886277211	3,79148298	-0,883922654
-7,360202642	9,112928105	9,477575509	-7,299057945
-1,008478023	6,285264488	2,072425747	9,72885151
-6,108961305	8,021328638	7,185082733	-6,101281821
-1,73341364	6,06432619	2,870195978	3,699739675
-5,555988201	7,280648898	6,556461319	-5,480176469
-6,360638592	8,222748739	7,510949859	-6,34162498
-0,128069518	6,205006491	1,293051968	16,49334556
-3,90708794	5,89323469	5,057663909	-3,906896028
0,992107173	6,656166746	0,118484489	32,89510543
-2,032330679	5,81947114	3,337257134	1,161540351
-0,755276119	6,075400998	1,952388214	10,26795309
-1,114403031	5,968202038	2,36729557	7,029765739
0,880758235	6,285965204	0,523604074	27,57578932
-0,939347187	6,00088868	2,18117343	8,433689122
-6,544094874	8,287386421	7,724002992	-6,478195644
-2,920911884	6,006711374	4,043139072	-1,741951352
-6,302332281	8,086345286	7,397080608	-6,255681899
-1,01454628	6,204165876	2,111760264	9,159126883
-0,844647419	5,820910977	2,309874831	8,013497346
-4,059425479	5,949571427	5,178347048	-4,047357566
-2,964400539	5,697812961	4,311825587	-2,426506759
-6,452963702	8,135153086	7,56353668	-6,351960114
-0,778943021	5,943767218	2,099824615	9,237169177
-3,086239425	6,129155225	4,156017161	-1,998566061
-5,964615859	7,880890786	7,007710132	-5,957605971
-0,119832393	6,27133601	1,239327953	17,11514989
-4,502470346	6,254832667	5,530416758	-4,441145926
-2,680624306	5,854618096	3,907438563	-1,302362888
-3,055830181	5,787654005	4,313933236	-2,52026407
-0,659397591	6,164329983	1,800356974	11,62515348
-0,996151454	6,126877328	2,136770409	8,805293046
-0,48992887	6,155958668	1,651487215	12,9498456
-0,248174354	6,127837287	1,461695836	14,80361012
-0,327588801	6,1232782	1,534047323	14,07966921
-7,18016276	8,574030393	8,547129589	-6,812766314
-1,695531264	5,977625847	2,891095653	3,512424426
-6,392459761	8,311124787	7,589702764	-6,385844383
-5,45608467	7,420231809	6,458499437	-5,454799242
0,05353715	6,152635974	1,188003773	17,7454295
-1,517532224	6,174716887	2,608036638	5,543098108
-6,040026949	7,777884667	7,065984076	-5,971308373
-6,128094993	8,097144435	7,22901752	-6,127137056
-4,478832958	6,196409702	5,516757486	-4,399070062
-0,127348346	6,114542802	1,373180364	15,77037128

-1,973568728	6,031269071	3,118137246	2,260561974
-7,234700159	8,967233796	9,098750184	-7,163161904
-5,959319722	7,771054642	6,984668301	-5,923875981
-1,589858246	5,718517321	3,066623475	2,941331209
-1,257688651	5,996052049	2,472456999	6,24094545
-0,183610851	6,161524139	1,376663494	15,64018328
-2,441129969	5,873920192	3,667334292	-0,388242146
0,570091304	6,321238661	0,63014411	24,49520013
-2,592757549	5,91044759	3,78380381	-0,856450504
-0,754361597	6,129933948	1,910721389	10,6401271
-6,033194698	7,846201263	7,069551943	-5,998228153

Таблиця №1. Фронт Парето для набору коефіцієнтів №1.



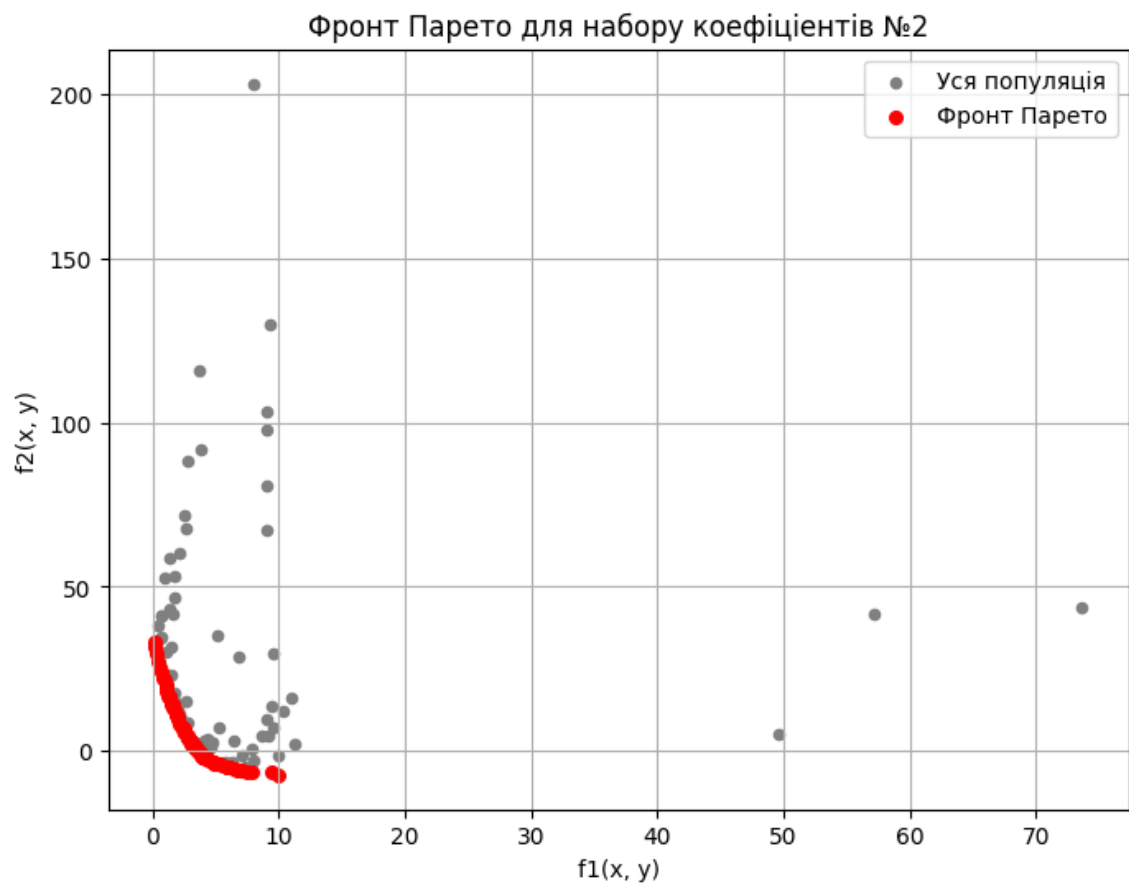
Коефіцієнти відбору/кросоверу/мутації: 10% - відбір, 80% - кросовер, 10% - мутація(№2):

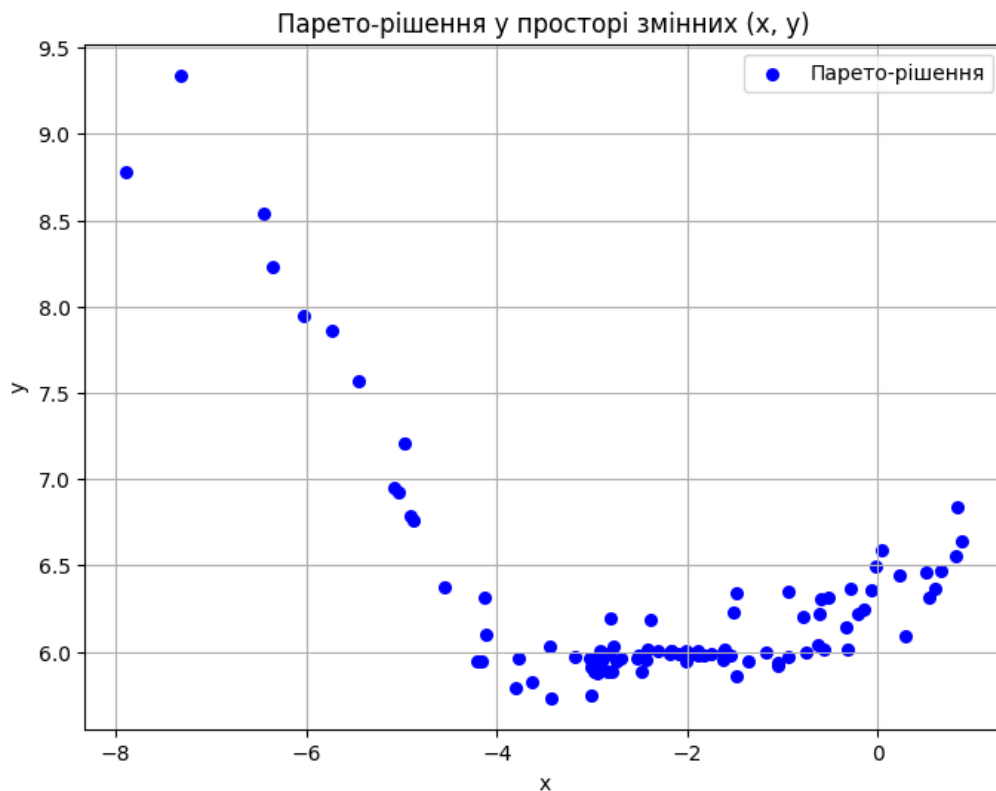
x	y	f1	f2
0,65717758	6,46514196	0,44650312	26,8953351
-6,03149575	7,94625854	7,08827786	-6,02423037
-2,30301238	6,00596504	3,44764315	0,59703538
0,59600239	6,3617176	0,57375294	25,1749899
-4,89942498	6,78724905	5,89959862	-4,88684154
-2,832884	5,8854075	4,02918725	-1,72507828
-2,96964091	5,88524787	4,15960135	-2,13130479
-2,94965139	5,94249931	4,10491878	-1,96390439
-2,70303097	5,95927	3,85818415	-1,12489446
-0,1400359	6,24917039	1,27180596	16,7449503
-4,12462596	6,31464758	5,14610695	-4,08851774
-0,75282919	5,99945385	2,01856299	9,7877425
0,8841312	6,63978676	0,17395853	31,3978008
-6,34698697	8,23070078	7,50148683	-6,33346449
-2,89373195	5,94712825	4,04845736	-1,78408817
-1,35288385	5,94675682	2,60127967	5,37529312
-1,9765502	5,98552927	3,1494448	2,05944672
-3,77694767	5,95848916	4,89856105	-3,74399036
-1,74477901	5,98623143	2,93087597	3,27932998
-0,52113802	6,31621199	1,59137641	13,8814484
0,29184138	6,0911127	1,08806727	19,5021279
-0,94088494	5,97336813	2,20859535	8,25506936
-2,7694469	6,03325867	3,88357903	-1,17222669
-0,62009504	6,03794736	1,86583547	11,0616195
-1,60634604	6,01104757	2,78380609	4,17624339
-2,82766511	5,99168515	3,96039098	-1,47272245
-1,47785769	6,34076819	2,51567968	6,71839882
-2,91999378	5,95150753	4,07122732	-1,85597316
-0,30973908	6,01188418	1,63362188	13,3961392
-3,42049873	5,73260179	4,70329704	-3,32309041
-0,21079576	6,224011	1,35226542	15,8951008
-2,09002024	5,98750881	3,2556311	1,5104426
-7,31033351	9,34043606	9,95320016	-7,30942735
-0,61317616	6,22396845	1,72192109	12,4246448
-0,94364438	6,34780294	1,98964456	10,6446511
0,03489426	6,58934305	0,97972866	21,4184649
0,5160511	6,45740834	0,56646382	25,2513499
-2,18675236	5,99107968	3,34537773	1,06884473
-5,07192524	6,94822626	6,07192583	-5,0566238
-2,87671029	5,95841707	4,02565272	-1,70662073
-3,00347564	5,74671154	4,30058323	-2,45107604

-3,62344313	5,821536	4,8275179	-3,58420235
-0,07036808	6,36060933	1,14578444	18,3358019
-4,16549162	5,94700811	5,28315467	-4,11775658
-1,87106274	5,98465079	3,05054523	2,59619173
-3,0100334	5,91076596	4,18186393	-2,19871426
-3,44399551	6,02792342	4,54334615	-3,10302371
-1,88389506	5,97675681	3,0680801	2,49617521
0,83481516	6,83796573	0,16725837	33,0152582
-2,89344513	5,99909566	4,02026583	-1,67098206
-1,16904654	5,99814407	2,39002245	6,83474631
-1,62166031	5,99608981	2,80870708	4,01625518
-7,88646493	8,78078675	9,43534724	-6,66394069
-1,54750946	5,97982642	2,75190364	4,36865633
-2,79648114	5,8855167	3,99449802	-1,61048268
-2,47952621	5,88883265	3,6920955	-0,49338159
-5,73009004	7,86242986	6,77106553	-5,71257621
-2,4108978	6,01546266	3,54595445	0,16373059
-1,52069355	6,22555726	2,59106347	5,79559414
-3,79671255	5,79462399	5,01193183	-3,79670818
-2,94696414	5,88538928	4,13787107	-2,0663224
-4,10474195	6,10279259	5,16782207	-4,10473815
-2,52055123	5,9646095	3,68015445	-0,43524694
-2,91575207	6,00310216	4,03989644	-1,73342185
-2,43037268	5,94931036	3,60363143	-0,123201
-2,73948074	5,94680127	3,90052586	-1,28185787
-2,37989979	6,18536057	3,44443569	0,87978882
0,82693102	6,55484318	0,26310089	29,7904245
-0,56410428	6,00979163	1,84603011	11,3086571
-2,16443062	6,00341781	3,31662778	1,21744325
-3,02264915	5,96094494	4,16501126	-2,14225015
-2,9364157	5,97041438	4,07664903	-1,86726244
-1,04162781	5,9219103	2,34928348	7,2543994
-2,51063744	5,97636543	3,66367551	-0,36227891
-2,94487913	5,8745175	4,14326293	-2,08065163
-2,80155107	6,1948744	3,8564218	-0,86020117
-5,45405799	7,56846592	6,46214305	-5,44096882
0,22337312	6,44135288	0,83698707	21,9830418
-1,05203607	5,93417248	2,34548959	7,25467425
-1,81769702	5,98171882	3,00242751	2,86529333
-6,45373911	8,54317004	7,82490689	-6,44574122
-2,92972799	5,88527801	4,12150817	-2,01665216
-3,17818428	5,97090074	4,31031331	-2,5497849
-0,28767947	6,36904357	1,34781557	16,3698535
-1,8926521	6,00305822	3,05863684	2,56116186
-0,32673614	6,14135653	1,51782535	14,2245926
-4,88152598	6,7614552	5,88180124	-4,86710899
-4,20768665	5,94709654	5,32437854	-4,13977945

-1,48867255	5,86278775	2,80463742	4,14775041
-4,54779003	6,37628	5,56141313	-4,51837434
-4,96745474	7,20552125	5,96760423	-4,91077908
-1,616779	5,95180358	2,83808185	3,83556082
-2,00905144	6,00247795	3,16930837	1,9646978
0,54227297	6,31776144	0,65280718	24,1622075
-0,0179331	6,49756243	1,04876857	20,0491459
-0,60205783	6,30282109	1,67416907	13,0935909
-2,00734957	5,94073409	3,20984965	1,73062614
-1,94879774	5,98522967	3,12342942	2,19825726
-0,78862158	6,19971179	1,89983097	10,8469148
-2,17440506	6,00485818	3,32529035	1,17615358
-5,03033169	6,92056333	6,03033499	-5,01828259

Таблиця №2. Фронт Парето для набору коефіцієнтів №2.





Коефіцієнти відбору/кросоверу/мутації 30% - відбір, 50% - кросовер, 20% - мутація(№3):

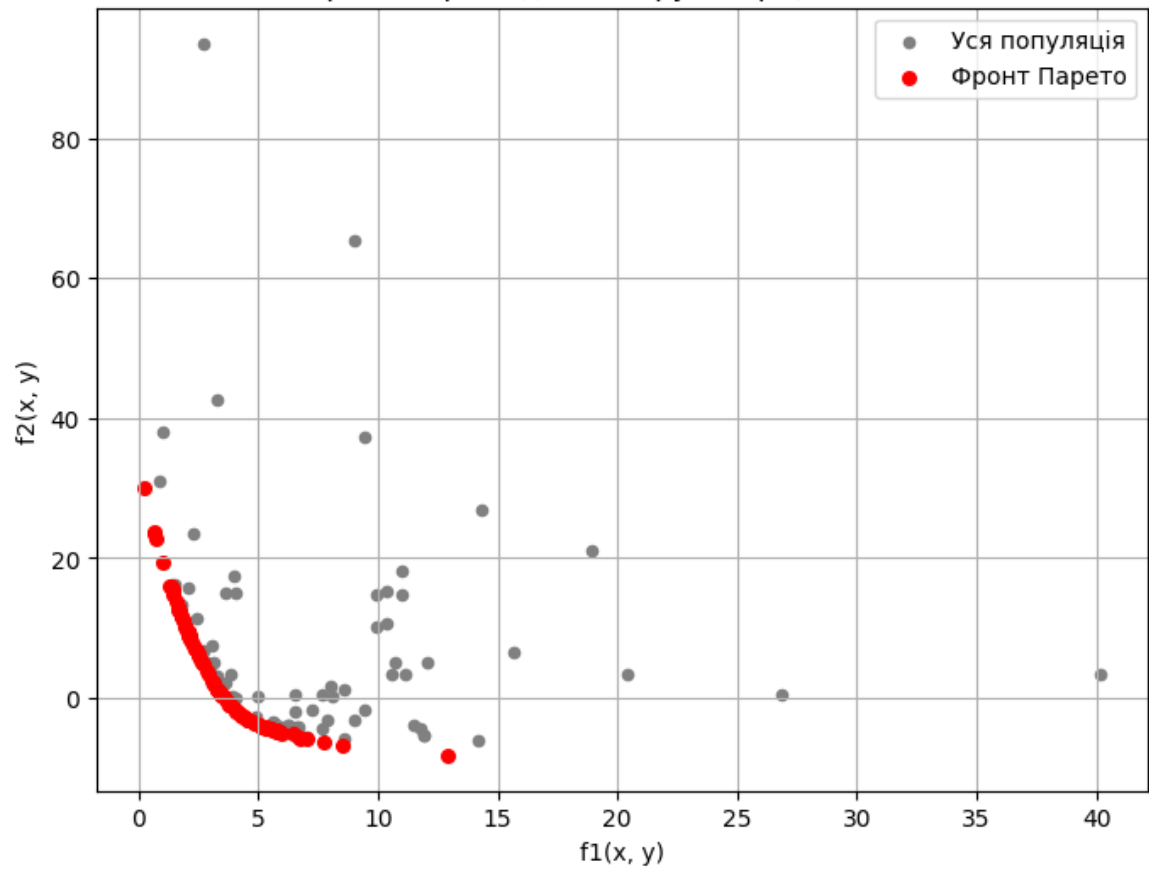
x	y	f1	f2
-7,15003043	8,60513767	8,54758316	-6,85312231
-4,70994027	6,70124529	5,71063781	-4,70986467
-0,49799342	6,13467312	1,6747154	12,7274458
-2,21463279	5,86897006	3,4598107	0,52219902
-1,91795524	5,95286279	3,11717291	2,22289352
-1,76797684	6,04462835	2,91458075	3,41516523
-0,36989902	6,04331379	1,64751423	13,1240771
-4,6471705	6,77552765	5,64739529	-4,63069494
-0,97856234	6,11174088	2,13007903	8,83824544
-8,27498318	10	12,9238273	-8,19936743
-3,7728069	5,98185666	4,88408212	-3,72910509
-0,60080036	6,10130212	1,7930067	11,6527122
-3,29441727	6,08756073	4,37437442	-2,66534072
-0,90327154	6,03686244	2,11729718	8,91612036
0,79891085	6,60956623	0,25233777	30,0505352
-5,99642296	7,80263073	7,0260194	-5,95886753
-2,55608102	6,00564355	3,69097933	-0,45484949
0,44939006	6,26000858	0,77654566	22,6278256
-4,70699173	6,73178869	5,7074451	-4,70637684
-4,350459	6,4843699	5,35706083	-4,33252687
-0,77781059	6,07940865	1,9694789	10,1227391

-0,13458176	6,12264124	1,37105936	15,7700366
-2,2847542	5,85426585	3,53734493	0,17861263
-0,85155733	6,04791557	2,06153897	9,36514865
-4,29554672	6,35078523	5,31229332	-4,29249543
-4,93090716	6,94200348	5,93090811	-4,93078403
-2,08641931	5,922696	3,29741401	1,28549277
-2,60785617	5,90816564	3,79970182	-0,91705144
-2,55929495	5,94874798	3,72691489	-0,62871521
-0,75683689	6,10482983	1,93095962	10,4522198
-0,12198953	6,05598618	1,43283997	15,3543401
-3,30092676	5,8693633	4,48688338	-2,97780665
-4,01802897	6,07933209	5,08911489	-4,0142709
-4,23255893	6,37299983	5,24730631	-4,21283529
-1,20032637	6,06634731	2,36670865	7,01374967
-6,60632196	8,15475223	7,72232024	-6,40240674
0,06269197	6,33855178	1,03439094	19,4336385
-3,20301019	6,07139257	4,29055663	-2,44892223
-2,0870779	5,92244013	3,29822453	1,28147665
0,38906536	6,42405795	0,69517779	23,5552213
-4,73293599	6,68673608	5,73377584	-4,73080156
-5,7750627	7,63418433	6,78698985	-5,75521598
-1,33847833	6,03851574	2,51457599	5,95172366
-0,93626752	6,01762442	2,16344081	8,55849287
-2,60581721	5,85457978	3,83708599	-1,04640924
-3,68337337	6,07576333	4,76063722	-3,52940349
-4,76990451	6,80432633	5,77003155	-4,76871965
-1,27470552	6,11460265	2,40599856	6,79031018
-0,49707051	6,07040904	1,72857078	12,2716777
-4,15577451	6,27897827	5,18191837	-4,14059534
-3,07454863	5,90513215	4,24722422	-2,38467965
-5,01101216	6,92918095	6,01101426	-5,00431582
-0,50282269	6,12108928	1,68973597	12,5890305
-5,6935914	7,66628006	6,70829627	-5,69284549
-4,81970298	6,70159364	5,82038419	-4,80575317
-0,74619117	6,19651791	1,86170932	11,1585635
-0,87423922	6,06900186	2,06495551	9,33226913
-4,74578903	6,71087348	5,7463971	-4,74456994
-0,60806189	6,04016325	1,85327602	11,1712579
-1,91682915	5,99961326	3,08373801	2,42116051
-2,1713438	5,91197356	3,38508606	0,85844814
-1,09498562	6,11892602	2,23418733	8,04922991
-0,8503215	6,05941348	2,05094833	9,44794983
-0,82942712	6,06766438	2,02543789	9,65675346
-0,96319879	6,14611626	2,09422103	9,16776484
-2,92690196	6,1373675	3,99678566	-1,46167513
-3,95512181	6,02890873	5,04405746	-3,94967731
-0,49406797	6,2814263	1,58077655	13,8500152

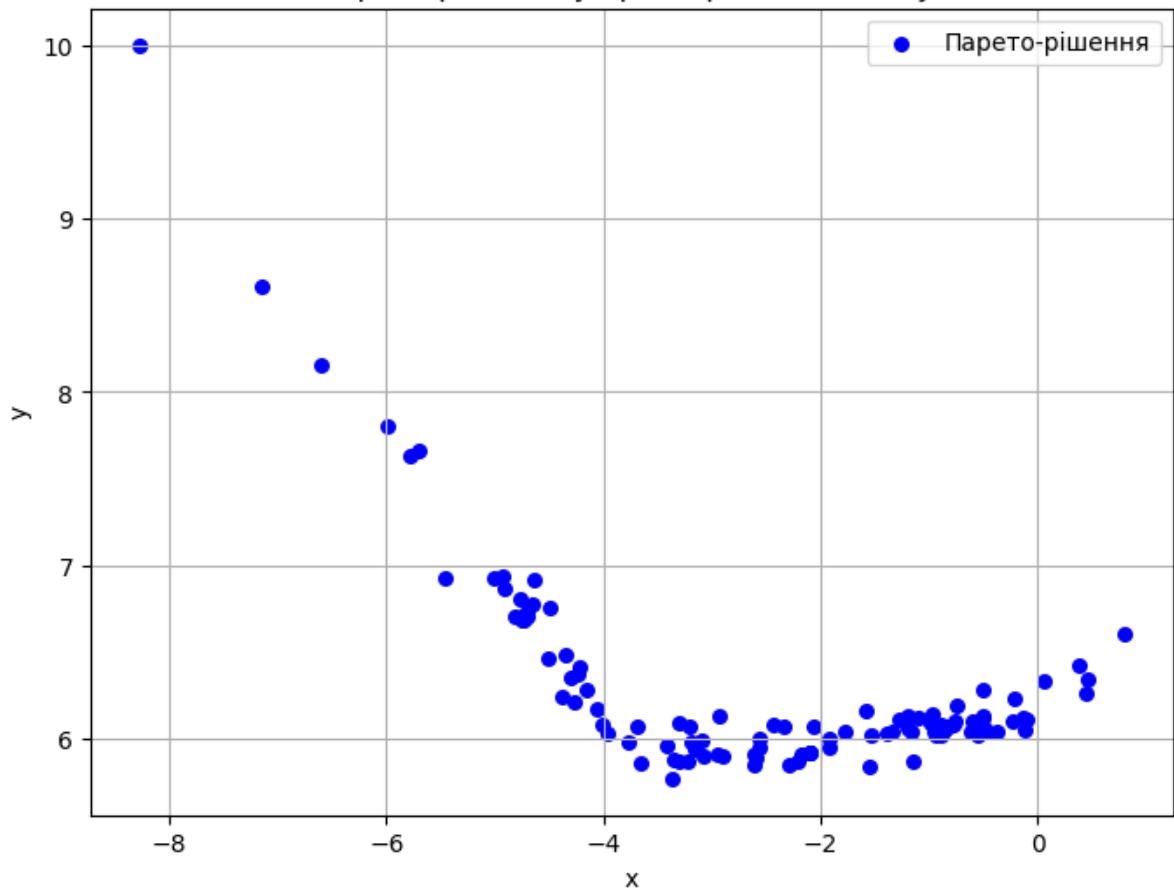
-0,88628089	6,02665736	2,11083268	8,97568347
-4,64239613	6,92119888	5,64239955	-4,56466516
-2,89676382	5,90254934	4,07864471	-1,88515931
-4,51398477	6,46913317	5,52118197	-4,51197311
-0,20310193	6,23211945	1,3398247	16,0298803
-3,66239135	5,86209239	4,83885188	-3,62251085
-4,06624061	6,16973314	5,1129236	-4,05552991
-0,10609994	6,11644206	1,35385006	15,9767439
-0,95856346	6,03913491	2,16526743	8,53135701
-1,18890256	6,13605274	2,31266397	7,49679166
-0,98365546	6,09550334	2,14573993	8,69994177
-3,18635911	5,98005268	4,31367722	-2,55640963
-3,15502165	5,93777925	4,3054959	-2,54231218
-1,52418193	6,02446297	2,69762395	4,72722332
-1,13835399	5,86884083	2,49193338	6,31720437
-5,4513817	6,92822451	6,45138375	-5,17768825
0,47257462	6,34717684	0,67808944	23,7025787
-4,38693389	6,2459685	5,41685529	-4,36706265
-1,16581498	6,04297126	2,35151753	7,11221329
-3,40938736	5,96041121	4,53990157	-3,10576007
-2,33946594	6,07275647	3,44836996	0,66483013
-4,4907966	6,76089714	5,49109422	-4,41784229
-3,22302771	5,8678908	4,41323512	-2,8071793
-3,35642908	5,88445006	4,53068806	-3,07762293
-0,54092395	6,02036916	1,81533076	11,565615
-2,42895934	6,07788822	3,53281127	0,29000712
-4,27056418	6,21420106	5,30661175	-4,26738738
-1,54389957	5,84054217	2,87726983	3,73066765
-0,88625988	6,08032158	2,06721266	9,31577027
-2,93984312	5,91180304	4,11395535	-1,99513705
-1,57698228	6,16090437	2,67143565	5,09967105
-3,36167163	5,77300524	4,61419097	-3,1924763
-2,05741345	6,07196084	3,17640364	2,00098773
-4,90467428	6,86613606	5,90470147	-4,90318908
-4,21245813	6,41885804	5,22338765	-4,16985721
-4,75607573	6,68258864	5,75695738	-4,75067537
-0,22563122	6,10541783	1,4637676	14,8271129
-0,44996233	6,03785868	1,72027344	12,423038
-3,0975162	5,9899572	4,22260813	-2,30106525
-2,08758779	5,92226699	3,29883316	1,27845998
-1,39285746	6,03504815	2,56763972	5,58831422
-2,5963043	5,89015613	3,80139725	-0,92225172

Таблиця №3. Фронт Парето для набору коефіцієнтів №3.

Фронт Парето для набору коефіцієнтів №3



Парето-рішення у просторі змінних (x, y)

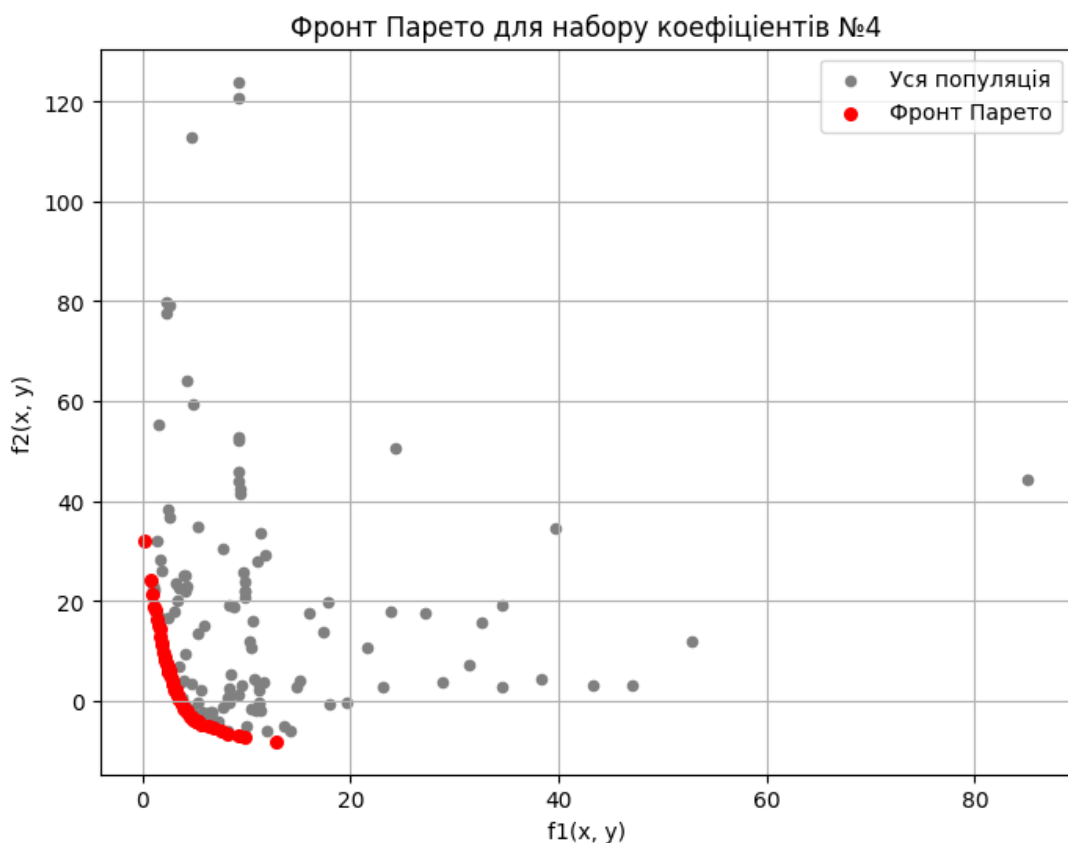


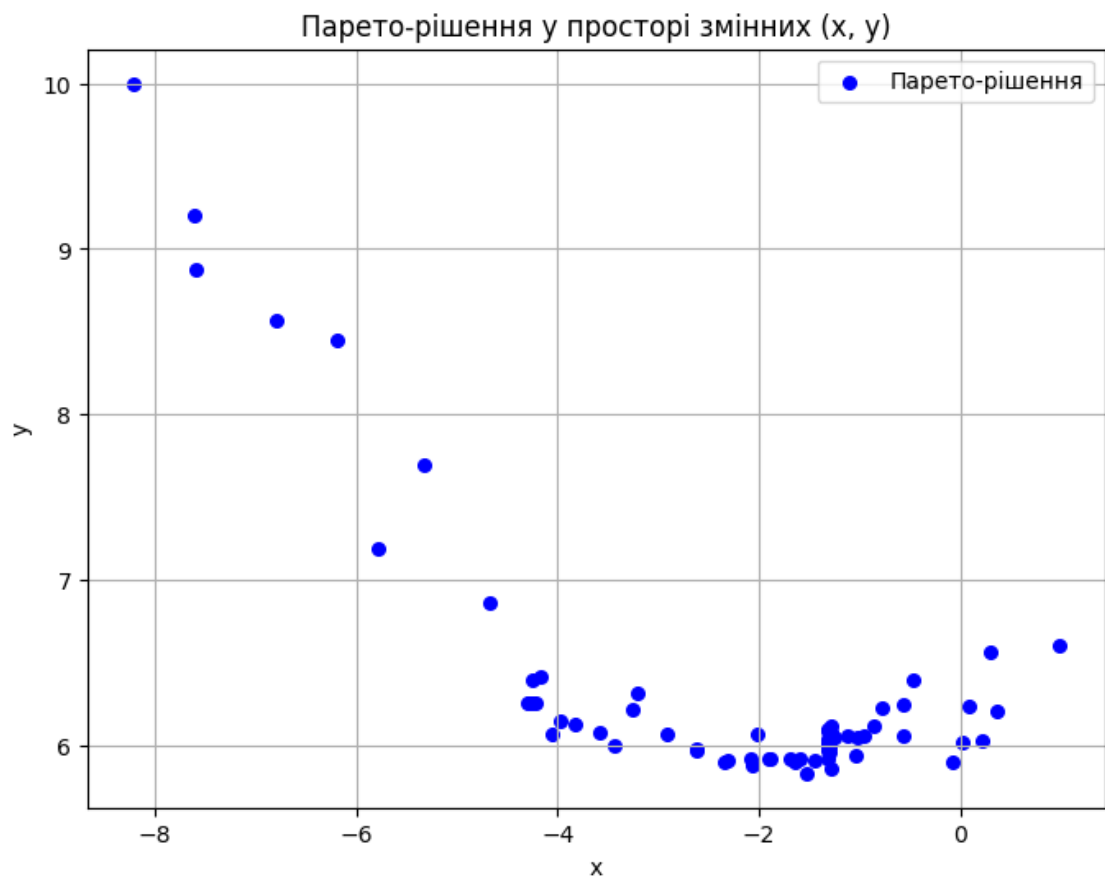
Коефіцієнти відбору/кросоверу/мутації 10% - відбір, 30% - кросовер, 60% - мутація(№4):

x	y	f1	f2
-7,601304512	8,879412786	9,298323645	-7,080176848
-7,608502126	9,202975698	9,88225213	-7,444050442
-0,569619903	6,05654347	1,804439761	11,58901606
-1,31600387	5,920722942	2,592435872	5,468557572
-1,909740137	5,920864051	3,134123167	2,13487926
-4,280482379	6,251520775	5,310116938	-4,279643604
-4,256830323	6,398840613	5,269238053	-4,2366634
-0,854871629	6,114229141	2,013983677	9,768539759
-1,591021797	5,92070165	2,840836294	3,836386421
-3,589488543	6,078604125	4,667349972	-3,350254491
-1,298034106	5,985050147	2,518355528	5,9220211
-3,821202445	6,125335795	4,881523876	-3,72870535
-1,695347483	5,92070292	2,936298446	3,256859338
-1,296649178	6,010865639	2,496364987	6,070321818
-2,618887825	5,968190781	3,77223895	-0,798269359
-1,285681392	6,120310986	2,413128161	6,749443542
-1,312344689	6,093424031	2,454062892	6,422057615
-4,212701448	6,250627118	5,242862358	-4,211263092
-1,297584793	6,009707376	2,498124922	6,058024114
-1,315506567	6,0418185	2,49088336	6,11727019
-2,91515847	6,06690913	4,010799446	-1,588628887
-0,951162859	6,057366377	2,143961772	8,697337439
-1,308522711	6,087073677	2,45436145	6,411822758
-2,059211189	5,876979309	3,308978191	1,245069751
-1,298910205	5,962142166	2,538747299	5,79389427
-0,073990738	5,897298941	1,622340778	14,54369487
-1,881459289	5,920863178	3,107885516	2,277708934
-1,278389795	5,85406485	2,629727855	5,355712198
-4,176317377	6,410904612	5,187937331	-4,121286206
-1,034668237	5,933400448	2,331112895	7,367980194
-1,114690971	6,061375369	2,29087482	7,568257972
-3,263958822	6,216165124	4,307995897	-2,35726198
-2,334856816	5,901193329	3,546692407	0,118553255
-6,789801945	8,572332441	8,172694298	-6,74250896
-1,308131468	5,964103826	2,545382772	5,746057697
-1,310454714	5,995430399	2,521230862	5,898639715
0,3711036	6,207348444	0,888969718	21,33332673
-1,317587076	6,003495704	2,521368713	5,896518085
-3,42978874	5,996447519	4,542831068	-3,108686569
0,017980445	6,0220163	1,370825685	16,33955414
-1,013485034	6,045702424	2,209856801	8,180857264
-1,638679149	5,900240441	2,90266222	3,47598033
-1,53535064	5,831230917	2,879933454	3,735715609

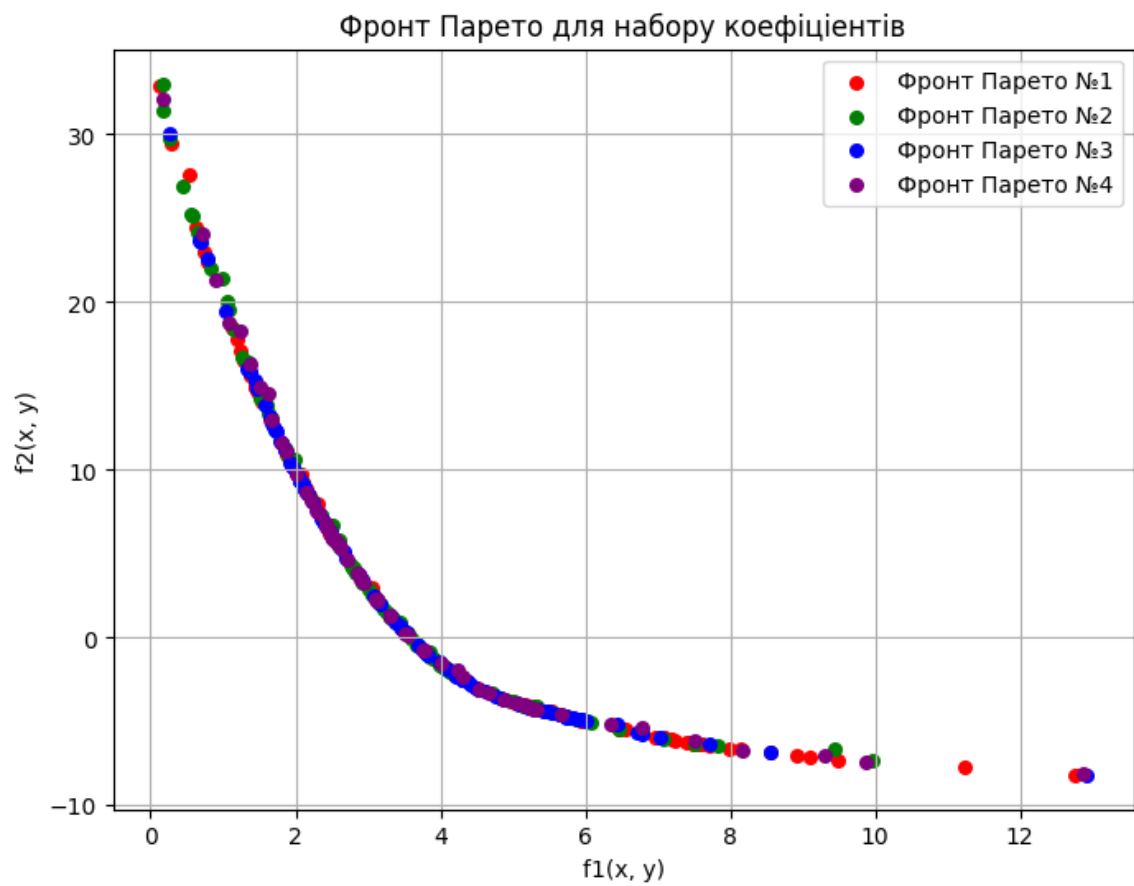
-4,307869941	6,25189957	5,337292965	-4,304737259
-2,011146804	6,070968698	3,13240247	2,231719431
-4,055427827	6,063912938	5,130806986	-4,05535583
-0,467423913	6,394706093	1,512470749	14,95612141
-1,321958976	6,002515592	2,526156968	5,863424792
-3,978585388	6,146414796	5,031618194	-3,950418678
-2,08328828	5,920814504	3,295915056	1,293214346
-2,304490151	5,903346332	3,516535286	0,251850937
-0,77323431	6,222520337	1,873432543	11,12433978
-4,257525333	6,251027776	5,287366837	-4,257483115
-1,288799084	6,0789381	2,44096446	6,496076644
-3,207670648	6,312499456	4,234134731	-1,987023953
-2,62349783	5,977410113	3,771366366	-0,790419362
-1,445813052	5,910278071	2,718849611	4,627774778
-1,242905351	6,047974523	2,419111259	6,625507706
0,21445167	6,029098114	1,227059938	18,22216644
0,305931502	6,564643502	0,719482246	24,02843237
-4,674274558	6,861861617	5,674306644	-4,639085653
0,09199382	6,231155966	1,083466398	18,78161789
-0,566442254	6,241035339	1,66899636	12,93619209
-5,784442086	7,183423867	6,784525507	-5,423219187
0,980721869	6,598491757	0,162357463	32,10834656
-8,215080663	10	12,88090492	-8,168820971
-6,201301063	8,448133719	7,500435239	-6,140374703
-5,333999516	7,694607358	6,352348874	-5,2039615

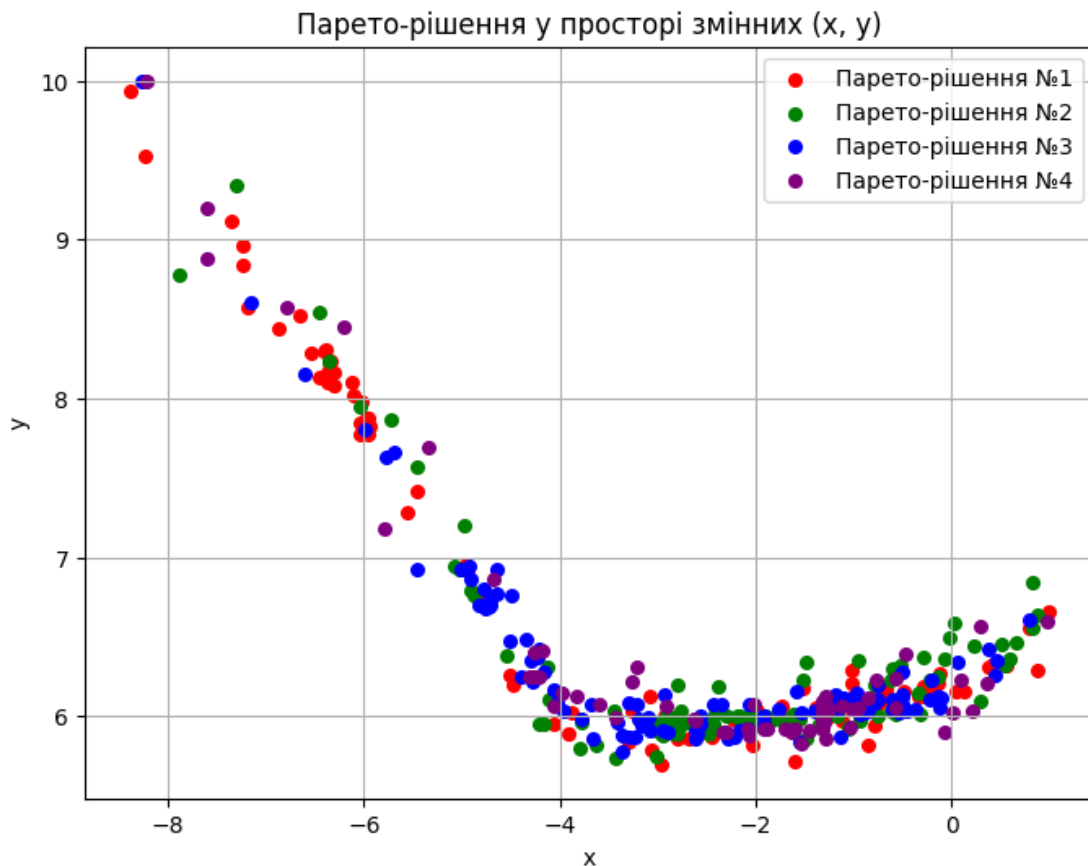
Таблиця №4. Фронт Парето для набору коефіцієнтів №4.





Фронти Парето для тестованих коефіцієнтів:





Висновок:

У межах лабораторної роботи № 2 було реалізовано генетичний алгоритм для задачі багатокритеріальної оптимізації з двома цільовими функціями (варіант №36) та проведено аналіз впливу різних комбінацій параметрів (відбір / кросовер / мутація) на формування Парето-оптимальної множини.

Було протестовано чотири набори коефіцієнтів:

1. Набір №1 (відбір 20%, кросовер 70%, мутація 10%) продемонстрував найкращий баланс між якістю та щільністю рішень. Фронт Парето вийшов плавним, рівномірно заповненим і стабільним. Більшість особин у популяції знаходяться поблизу ефективної множини, що свідчить про ефективність цього налаштування.
2. Набір №2 (відбір 10%, кросовер 80%, мутація 10%) сформував фронт із менш рівномірним заповненням. Спостерігається зміщення рішень до нижніх значень функцій і зменшення різноманіття, що може свідчити про передчасну збіжність до часткового рішення без достатньої експлорації простору.
3. Набір №3 (відбір 30%, кросовер 50%, мутація 20%) показав ширше охоплення Парето-фронт. Хоча щільність точок дещо нижча, фронт

охоплює різні області компромісу між критеріями, забезпечуючи хорошу різноманітність рішень.

4. Набір №4 (відбір 10%, кросовер 30%, мутація 60%) дав неочікувано щільний і компактний фронт, незважаючи на високу стохастичність через мутацію. Це може свідчити про те, що навіть при агресивному випадковому пошуку можна досягти якісної множини компромісних рішень за умови правильного налаштування відбору.

Експериментально доведено, що параметри генетичного алгоритму мають суттєвий вплив на результат багатокритеріальної оптимізації. Найбільш ефективним для стабільного формування фронту Парето виявився класичний набір параметрів (20/70/10), однак для ширшого покриття простору рішень доцільно застосовувати більшу частку мутації (як у наборі №3 або №4).

Варто також зазначити, що результати можуть мати певну варіативність через стохастичну природу генетичних алгоритмів, тобто елемент випадковості може впливати на остаточну форму та якість Парето-оптимальної множини.

Додаток А

```
import numpy as np
import random
import copy
import matplotlib.pyplot as plt
import pandas as pd
import os

# === Цільові функції для багатокритеріальної оптимізації (варіант 36) ===

def f1(x, y):
    # f1 = sqrt((x - 1)^2 + (y - 7)^4)
    return np.sqrt((x - 1)**2 + (y - 7)**4)

def f2(x, y):
    # f2 = (x + y - 2)^2 + x
```

```
return (x + y - 2)**2 + x
```

```
# === Параметри алгоритму ===
```

```
gen = 40          # Кількість поколінь
pop_size = 200     # Розмір популяції
crossover_rate = 0.7  # Ймовірність кросоверу
mutation_rate = 0.1   # Ймовірність мутації
selection_rate = 0.2  # Частка еліти (кращих рішень)
```

```
# === Генерація початкової популяції ===
```

```
def generate_population(size, lower, upper):
    return [[random.uniform(lower, upper), random.uniform(lower, upper)] for _ in
range(size)]
```

```
# === Парето-домінування ===
```

```
def dominates(a, b):
    # Повертає True, якщо a Парето-домінує b
    return all(x <= y for x, y in zip(a, b)) and any(x < y for x, y in zip(a, b))
```

```
# === Оцінка популяції: обчислення f1, f2 + Парето-фітнес ===
```

```
def evaluate_population(pop):
    fitness = [0] * len(pop)
    objectives = [(f1(ind[0], ind[1]), f2(ind[0], ind[1])) for ind in pop]
    for i in range(len(pop)):
        for j in range(len(pop)):
```

```
        if i != j and dominates(objectives[j], objectives[i]):
            fitness[i] += 1
    return fitness, objectives
```

```
# === Кросовер (лінійна комбінація) ===
```

```
def crossover(p1, p2, lower=-10, upper=10):
    beta1, beta2 = random.uniform(-1, 1), random.uniform(-1, 1)
    child_x = beta1 * p1[0] + (1 - beta1) * p2[0]
    child_y = beta2 * p1[1] + (1 - beta2) * p2[1]
    child_x = np.clip(child_x, lower, upper)
    child_y = np.clip(child_y, lower, upper)
    return [child_x, child_y]
```

```
# === Мутація (додавання шуму) ===
```

```
def mutate(p, lower, upper):
    p_mut = p.copy()
    p_mut[0] += np.random.normal(0, (upper - lower) / 6)
    p_mut[1] += np.random.normal(0, (upper - lower) / 6)
    # Обмеження межами області пошуку
    p_mut[0] = np.clip(p_mut[0], lower, upper)
    p_mut[1] = np.clip(p_mut[1], lower, upper)
    return p_mut
```

```
# === Основний алгоритм ===
```

```
def multiobjective_genetic_algorithm(lower, upper):
```

```
population = generate_population(pop_size, lower, upper)
```

```
for gen_num in range(gen):
```

```
    fitness, objectives = evaluate_population(population)
```

```
    # Відбір еліти
```

```
    elite_count = int(pop_size * selection_rate)
```

```
    elite_indices = np.argsort(fitness)[:elite_count]
```

```
    elite = [population[i] for i in elite_indices]
```

```
    new_population = copy.deepcopy(elite)
```

```
    # Кросовер
```

```
    while len(new_population) < pop_size * (1 - mutation_rate):
```

```
        p1, p2 = random.sample(elite, 2)
```

```
        child = crossover(p1, p2)
```

```
        new_population.append(child)
```

```
    # Мутація
```

```
    while len(new_population) < pop_size:
```

```
        ind = random.choice(elite)
```

```
        new_population.append(mutate(ind, lower, upper))
```

```
    population = new_population
```

```
# Фінальна оцінка і побудова фронту Парето
```

```
fitness, objectives = evaluate_population(population)
```

```
pareto_front = [objectives[i] for i in range(len(fitness)) if fitness[i] == 0]
```

```

return population, pareto_front, objectives

# === Запуск ===

population1, pareto1, all_objs1 = multiobjective_genetic_algorithm(-10, 10)

# Фінальна оцінка і побудова фронту Парето
fitness1, objectives1 = evaluate_population(population1)

# Збираємо точки фронту Парето
pareto_front1 = []
pareto_coords1 = []

for i in range(len(fitness1)):
    if fitness1[i] == 0:
        pareto_front1.append(objectives1[i])    # (f1, f2)
        pareto_coords1.append(population1[i])   # (x, y)

# === Вивід точок фронту Парето: x, y, f1, f2 ===
print("=== Фронт Парето (x, y, f1, f2) ===")
for (x, y), (f1_val, f2_val) in zip(pareto_coords1, pareto_front1):
    print(f"x = {x:.4f}, y = {y:.4f}, f1 = {f1_val:.4f}, f2 = {f2_val:.4f}")

# === Формування таблиці з результатами фронту Парето ===
pareto_table1 = {
    "x": [x for x, y in pareto_coords1],
    "y": [y for x, y in pareto_coords1],
    "f1": [f1_val for f1_val, f2_val in pareto_front1],

```

```

    "f2": [f2_val for f1_val, f2_val in pareto_front1]
}

# === Створення DataFrame і збереження у файл Excel ===
df_pareto = pd.DataFrame(pareto_table1)
file_path = os.path.abspath("rez1.xlsx")
df_pareto.to_excel(file_path, index=False)
print(f"\nФайл збережено за шляхом: {file_path}")

# Візуалізація фронту Парето
pareto_front1 = np.array(pareto_front1)
all_objs1 = np.array(objectives1)

plt.figure(figsize=(8, 6))
plt.scatter(all_objs1[:, 0], all_objs1[:, 1], c='gray', s=20, label='Уся популяція')
plt.scatter(pareto_front1[:, 0], pareto_front1[:, 1], c='red', s=30, label='Фронт Парето')
plt.xlabel("f1(x, y)")
plt.ylabel("f2(x, y)")
plt.title("Фронт Парето для набору коефіцієнтів №1")
plt.legend()
plt.grid(True)
plt.show()

# Візуалізація Парето-рішень у просторі змінних (x, y) — без підписів
pareto_coords1 = np.array(pareto_coords1)

plt.figure(figsize=(8, 6))

```

```
plt.scatter(pareto_coords1[:, 0], pareto_coords1[:, 1], c='blue', s=30, label='Парето-  
рішення')  
plt.xlabel("x")  
plt.ylabel("y")  
plt.title("Парето-рішення у просторі змінних (x, y)")  
plt.legend()  
plt.grid(True)  
plt.show()
```