# АНОТАЦІЯ

Розроблений web-додаток призначений для оптимізації режимів токарної обробки титану з субмікро- та нанокристалічною структурою за допомогою генетичного алгоритму.

Розроблене програмне забезпечення виконує усі етапи роботи генетичного алгоритму: ініціалізацію ГА, оцінку пристосованості особин, селекцію найкращих особин популяції для схрещення, застосування генетичних операторів, формування нової популяції та вибір найкращого результату.

Реалізований генетичний алгоритм багатокритеріальної оптимізації з використанням методу Шаффера на ПЕОМ, проведений аналіз коректності побудованих алгоритмів.

Web-додаток розроблений на ПЕОМ мовами PHP, HTML, з використанням каскадних таблиць стилю CSS. Програмне забезпечення розгорнуте на сервері Apache.

Вимоги до апаратного забезпечення: ПЕОМ на базі процесора Intel Pentium 4 і вище з ОЗУ не менше 512МБ, операційна система MS Windows XP і вище, з розгорнутим веб-сервером Apache.

Сумарній об’єм проекту програми 1 МБ. Результатом роботи програми є вивід на екран мінімумів обох функцій, а також значення параметрів, необхідних для отримання цих мінімумів.

Назва програмного продукту – «Genetic». Програма орієнтована на використання у операційних системах MS Windows від версії XP або Linux, з розгорнутим сервером Apache. Середа розробки: Notepad ++, для розгортання додатку був використаний FTP клієнт SmartFTP.

### ЗМІСТ

|  |  |
| --- | --- |
| 1. Призначення і умови застосування. Характеристики програми……... | 4 |
| 1. Структура програми………...…………………………………............... | 5 |
| 1. Виклик програми………………………………………………………... | 9 |
| 1. Перевірка програми…..…………………………………………............. | 10 |

# ПРИЗНАЧЕННЯ І УМОВИ ЗАСТОСУВАННЯ. ХАРАКТЕРИСТИКИ ПРОГРАМИ

Програма призначена для оптимізації режимів токарної обробки титану з субмікро- та нанокристалічною структурою за допомогою генетичного алгоритму.

Результатом роботи програми є вивід результатів роботи генетичного алгоритму на екран - мінімумів обох функцій, а також значення параметрів, необхідних для отримання цих мінімумів..

Вимоги до апаратного забезпечення: ПЕОМ на базі процесора Intel Pentium 4 і вище з ОЗУ не менше 512МБ, операційна система MS Windows XP і вище, з розгорнутим веб-сервером Apache.

Режим роботи – графічний. Час роботи програми не визначений – працює в режимі обробки повідомлень від користувача.

# СТРУКТУРА ПРОГРАМИ

Web-додаток складається з файлу index.php, у якому послідовно виконуються етапи роботи генетичного алгоритму. Алгоритм працює у циклі до тих пір, доки не буде виконана умова завершення алгоритму.

Реалізовані у програмі етапи роботи генетичного алгоритму:

|  |  |
| --- | --- |
| **Етап** | **Опис** |
| **Ініціалізація генетичного алгоритму** | На цьому етапі виконуються підготовка до виконання генетичного алгоритму.  Додаток зчитує введені користувачем параметри генетичного алгоритму – точність результату по кожній змінній q[i], а також розмір популяції popsize.  За допомогою введеного користувачем значення q, а також мінімальних та максимальних границь значень змінних s, v, l, d, D0, для кожного параметру вираховується необхідна довжина хромосоми L та крок квантування h.  Далі за допомогою функції rand() для кожної особини початкової популяції розміром popsize присвоюються початкові значення хромосом. |
| **Оцінка пристосованості особин** | На цьому етапі визначаються значення цільових функцій для кожної особини популяції. Для першої половини популяції використовується перша фітнес функція, для другої – друга. Фітнес функціями виступають функції продуктивності    та розміру зерна металу    отримані значення функцій виводяться на екран. |
| **Перевірка умови завершення роботи алгоритму** | Може бути декілька різноманітних умов завершення генетичного алгоритму – за часом роботи алгоритму, кількістю ітерацій або зупинка покращення результату. У розробленому додатку алгоритм завершується після п’ятдесяти ітерацій без покращення результату |
| **Селекція найкращих особин** | Проводиться за так званим турнірним відбором. Окремо для кожної з під популяцій за допомогою функції rand() випадково видираються дві особини та зрівнюються значення їх цільових функцій fitness[i] або fitness1[i]. Так як поставлена задача мінімізації функції, до подальшого схрещення допускається функція з меншим значенням фітнес-функції. Половина обраних для селекції особин стає першою підгрупою батьків mom[i], друга – dad[i]. |
| **Використання генетичних операторів** | Спочатку проводиться кодування хромосом кожної особини у зручну для схрещування бінарну форму. Для цього десятинне значення хромосоми ind[i] спочатку переводиться у бінарну форму за допомогою методу decbin(). Далі хромосоми chrom[i] нарощуються до необхідної довжини L за допомогою методу str\_pad().  Для виконання кроссоверу необхідно розбити кожну хромосому батьківських особин на дві частини. Точка розриву split обирається випадковим образом за допомогою функції rand(). Очевидно, що значення точки розриву не може бути більше, ніж зменшена на одиницю довжина найкоротшої хромосоми у особині.  Визначивши точки розриву, кожна хромосома розділяється на дві частини за допомого функції substr(). Далі частини хромосом батьків хрест-навхрест об’єднуються у хромосоми нових особин нащадків nextgen[i]: cross1mom об’єднується з cross2dad, а cross1dad з cross2mom. |
| **Формування нової популяції** | Отримана популяція нащадків nextgen[i] замінює собою батьківську популяцію ind[i]. Для вирахування значень цільових функцій на наступних ітераціях алгоритму, бінарні значення хромосом переводяться назад у десятинні значення за допомогою функції bindec(). |
| **Вибір найкращої особини** | Якщо була виконана умова завершення роботи генетичного алгоритму, з отриманої фінальної популяції обираються особини з найкращим значенням цільових функцій. Найкращі значення фітнес функцій fitness та fitness1, а також кращі значення параметрів s ,v ,l ,d ,D0 для першої функції, та значення s1, s2 для другої функції виводяться на екран. |

# ВИКЛИК ПРОГРАМИ

Для виклику web-додатку необхідно вказати шлях до сторінки index.php у адресній строчці web-браузеру.

# Перевірка програми

Для коректного результату розрахунків за допомогою генетичного алгоритму необхідно правильно підібрати розмір початкової популяції, а також довжину хромосом особин.

Занадто низькі значення цих множин не дадуть точного результату роботи алгоритму, але занадто великі будуть суттєво збільшувати час роботи програми, не покращуючи при цьому результат.

Спочатку знайдемо оптимальний розмір популяції. Для цього зафіксуємо значення коефіцієнтів q, зробивши їх достатньо великими, щоб забезпечувати точність результату:

Точність qs = 5; довжина хромосоми Ls = 10;

Точність qv = 1; довжина хромосоми Lv = 10;

Точність ql = 4; довжина хромосоми Ll = 12;

Точність qd = 4; довжина хромосоми Ld = 10;

Точність qd0 = 10; довжина хромосоми Ld0 = 12;

Далі проведемо розрахунки з популяціями розміром у 2000, 1000, 500, 200 та 100 особин. Результати експериментів наведені на рис. 4.1 – 4.5

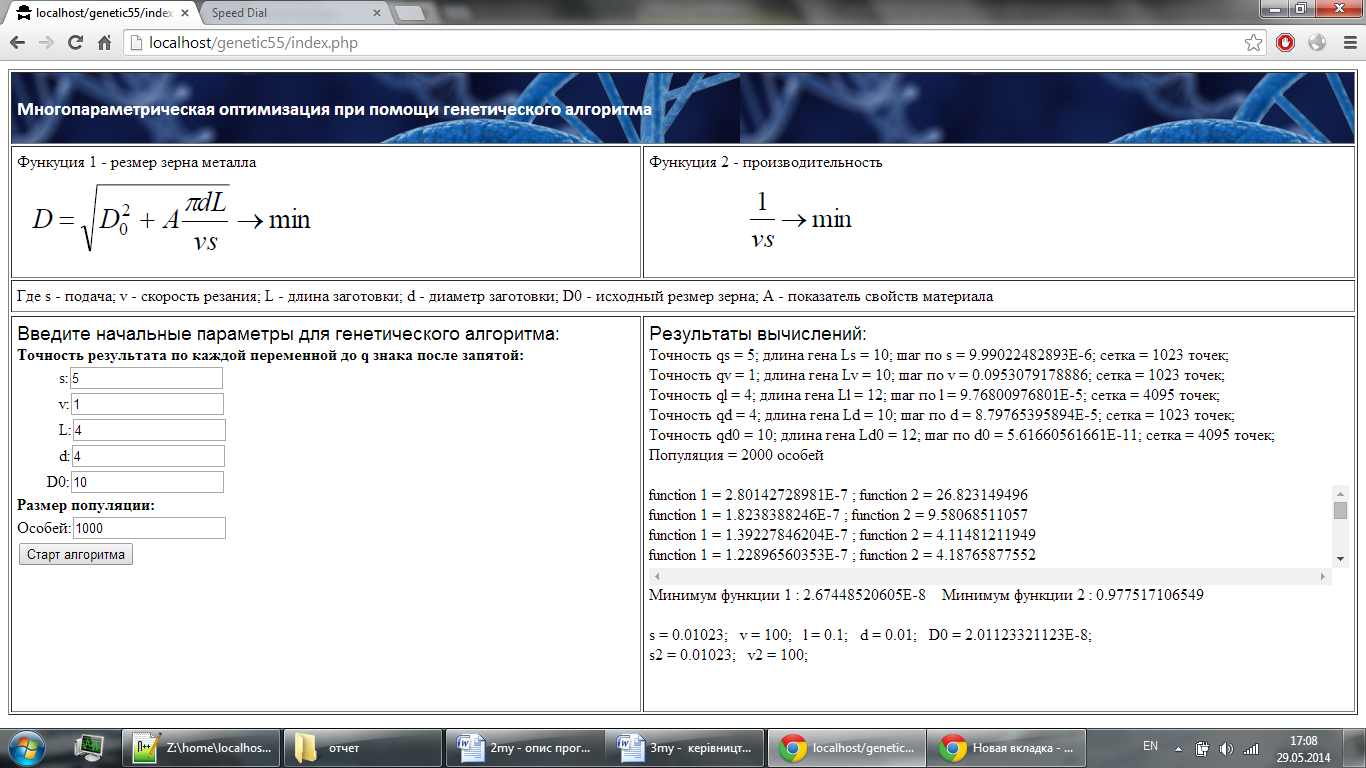


Рисунок 4.1 – розрахунок з популяцією у 2000 особин

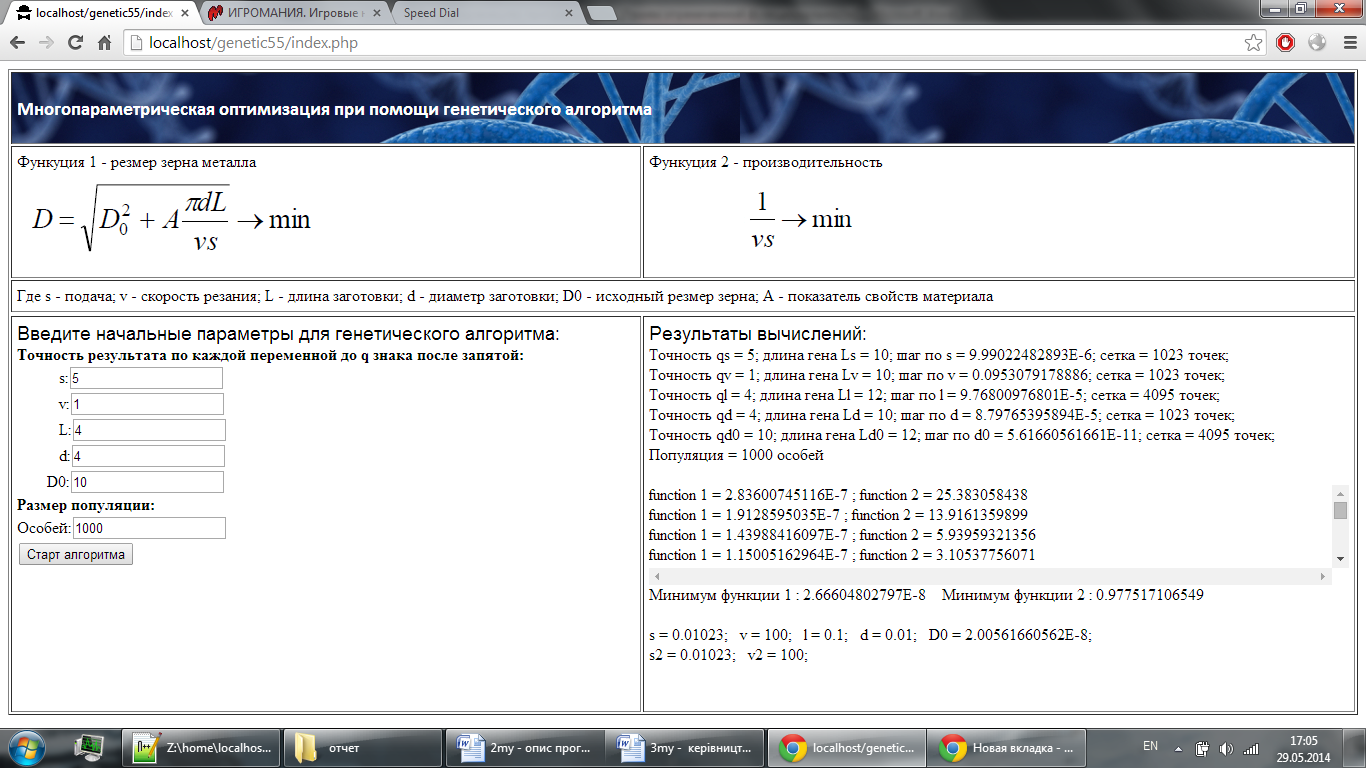


Рисунок 4.2 – розрахунок з популяцією у 1000 особин

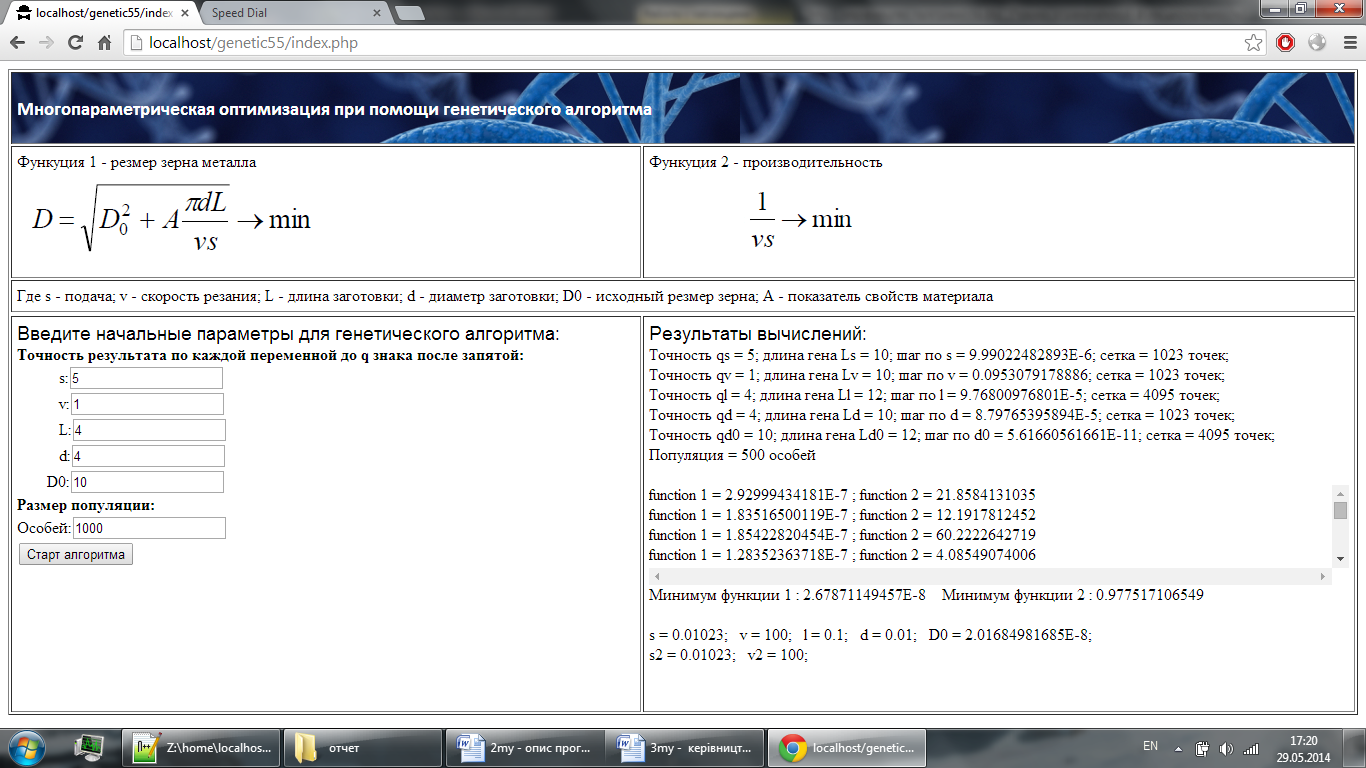


Рисунок 4.3 – розрахунок з популяцією у 500 особин

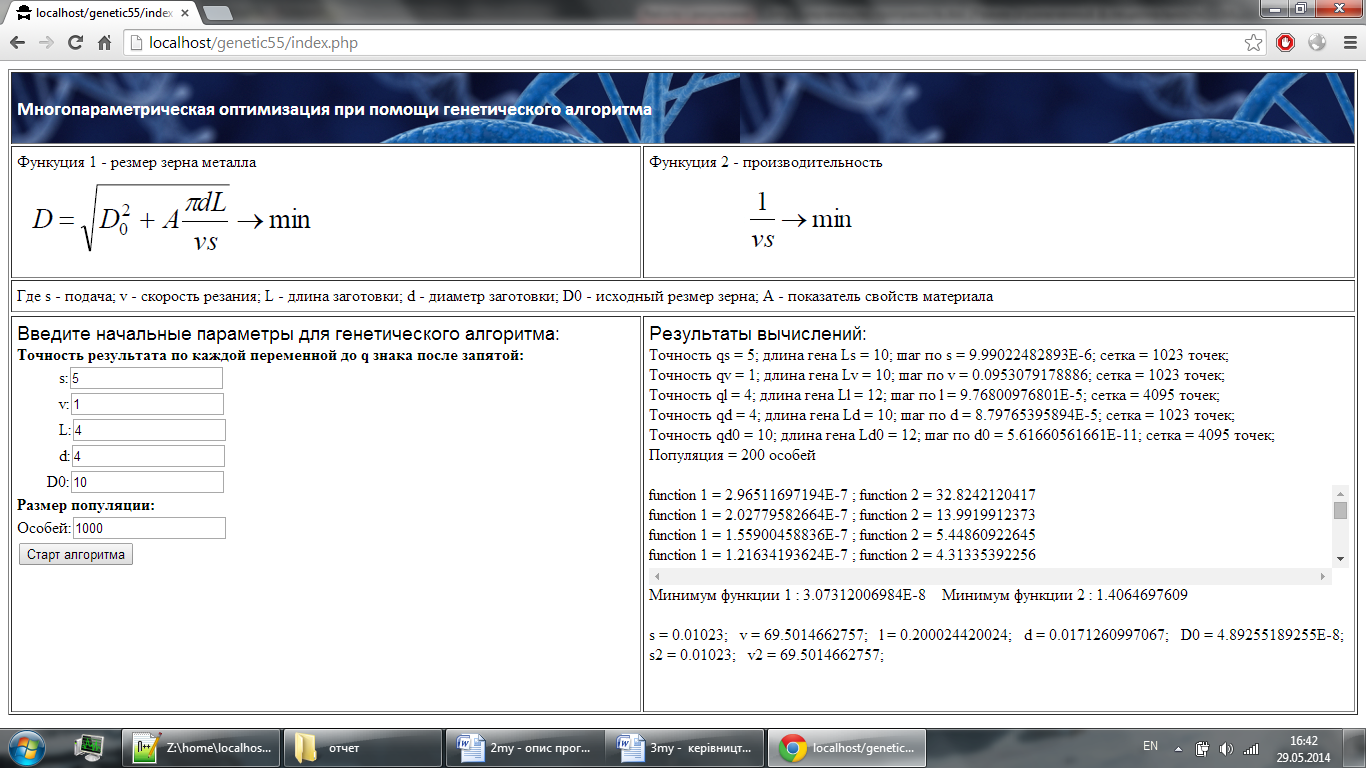


Рисунок 4.4 – розрахунок з популяцією у 200 особин

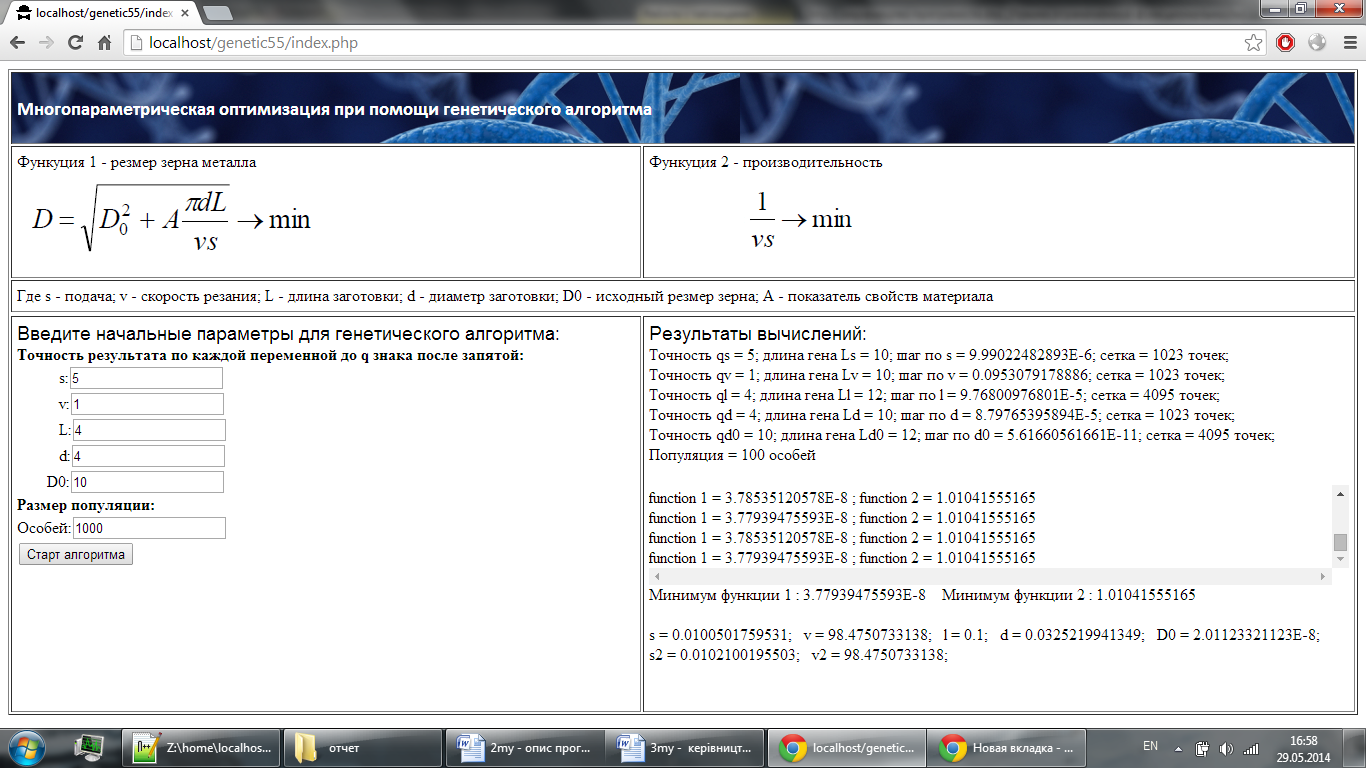


Рисунок 4.5 – розрахунок з популяцією у 100 особин

Результати, отримані під час розрахунків з різними розмірами популяції, наведені у таблиці 4.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| популяція | s | v | l | d | d0 | F1 | F2 |
| 2000 | 0.0123 | 100 | 0.1 | 0.01 | 2.011233211E-8 | 2.674485206E-8 | 0.97751710 |
| 1000 | 0.0123 | 100 | 0.1 | 0.01 | 2.005616605E-8 | 2.666048027E-8 | 0.97751710 |
| 500 | 0.0123 | 100 | 0.1 | 0.01 | 2.011233211E-8 | 2.857359789E-8 | 0.97751710 |
| 200 | 0.01023 | 69.5 | 0.2 | 0.017 | 4.892551892E-8 | 3.073120069E-8 | 1.40646976 |
| 100 | 0.01 | 98.47 | 0.1 | 0.032 | 2.011233211E-8 | 3.779394755E-8 | 1.01041555 |

Таблиця 4.1 – результати розрахунків

Як можна побачити, розрахунки з популяціями розміром у 100 та 200 особин дають результат, який суттєво відрізняється від інших, та є значно гіршим. Це означає, що таких розмірів популяції недостатньо для отримання правильного результату. Популяції розміром у 500 та 1000 особин дають приблизно однакові результати з незначними відхиленнями. Але збільшення популяції до 2000 особин не дає покращення результату, тобто така популяція є занадто великою. Отже, оптимальним розміром популяції для багатокритеріальної оптимізації заданих функцій буде 500 – 1000 особин.

Далі підберемо оптимальні значення коефіцієнтів q, які відповідають за шаг квантування та точність отриманих розрахунків. Процес аналізу наведений на рисунках 4.6 – 4.8.

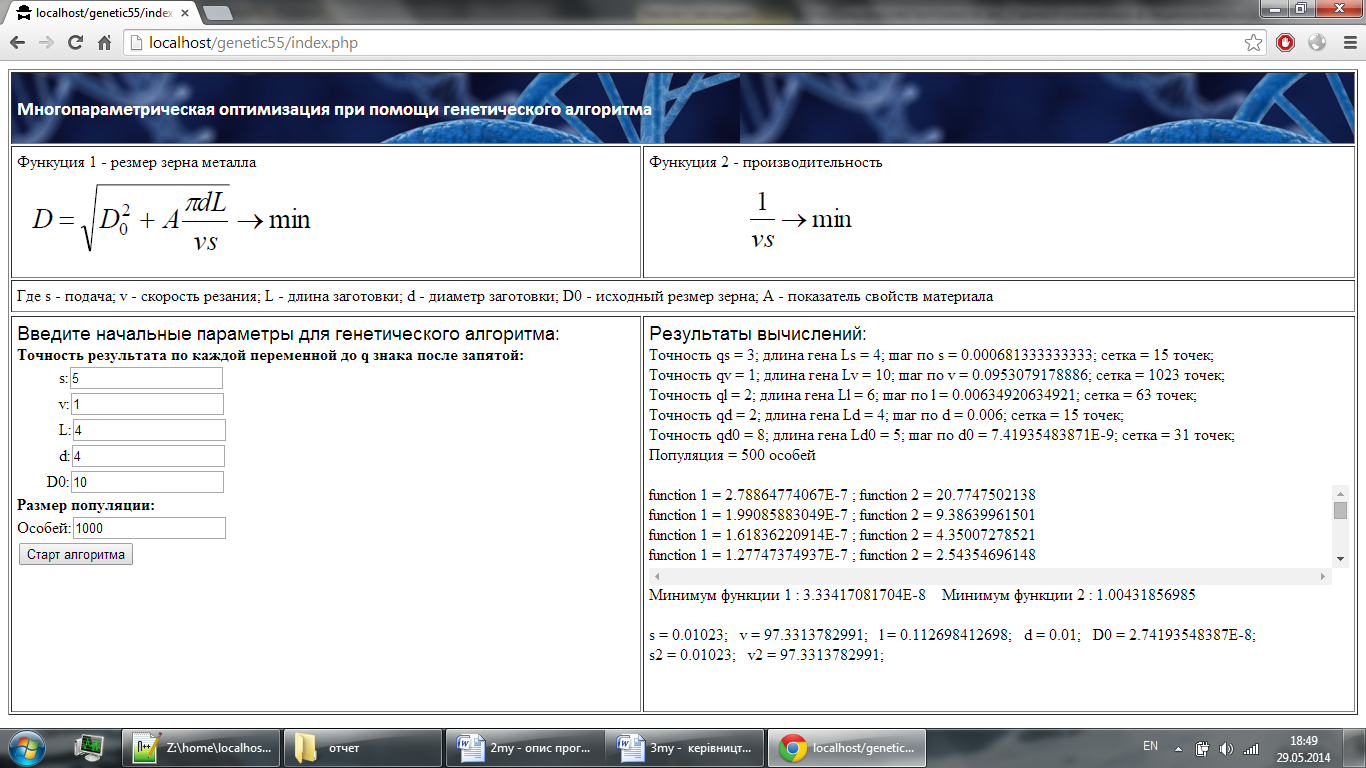


Рисунок 4.6 – розрахунок зі значеннями q = (3, 1, 2, 2, 8)

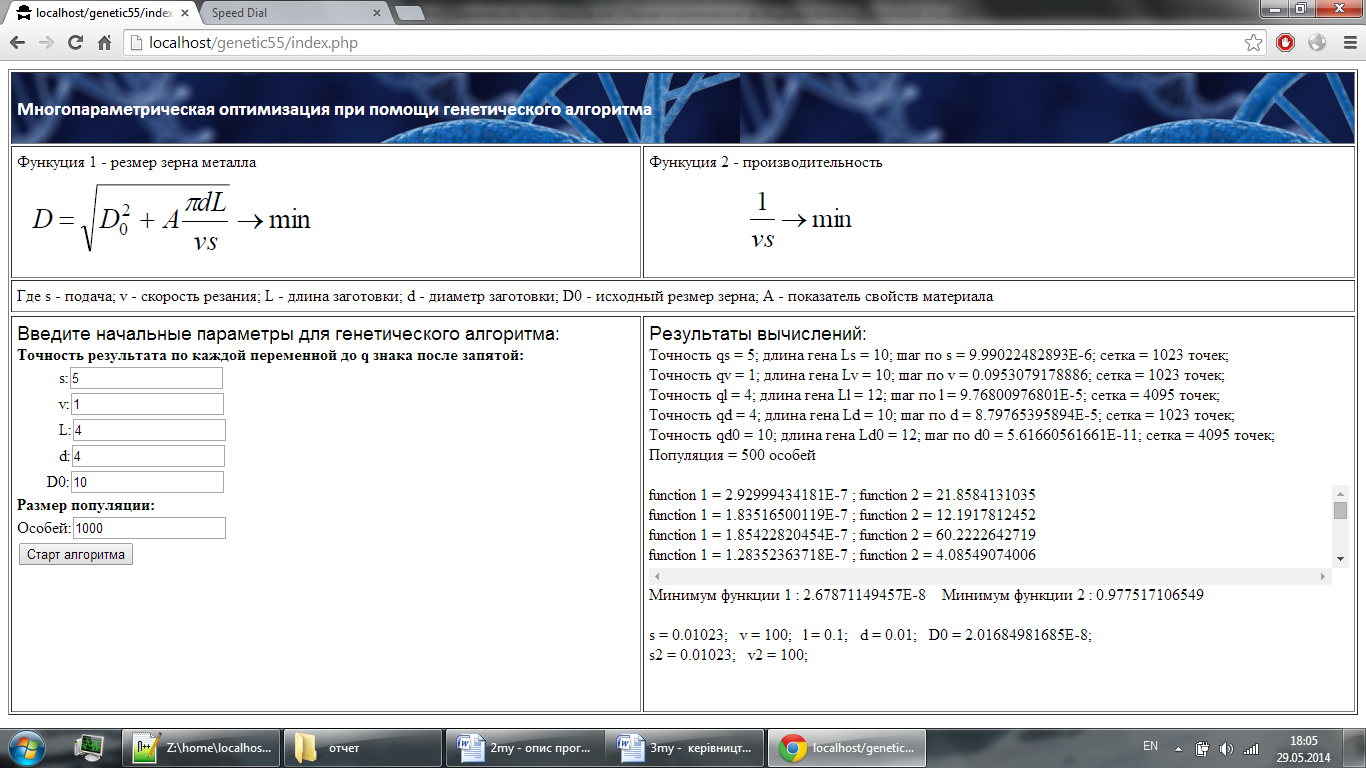


Рисунок 4.7 – розрахунок зі значеннями q = (5, 1, 4, 4, 9)

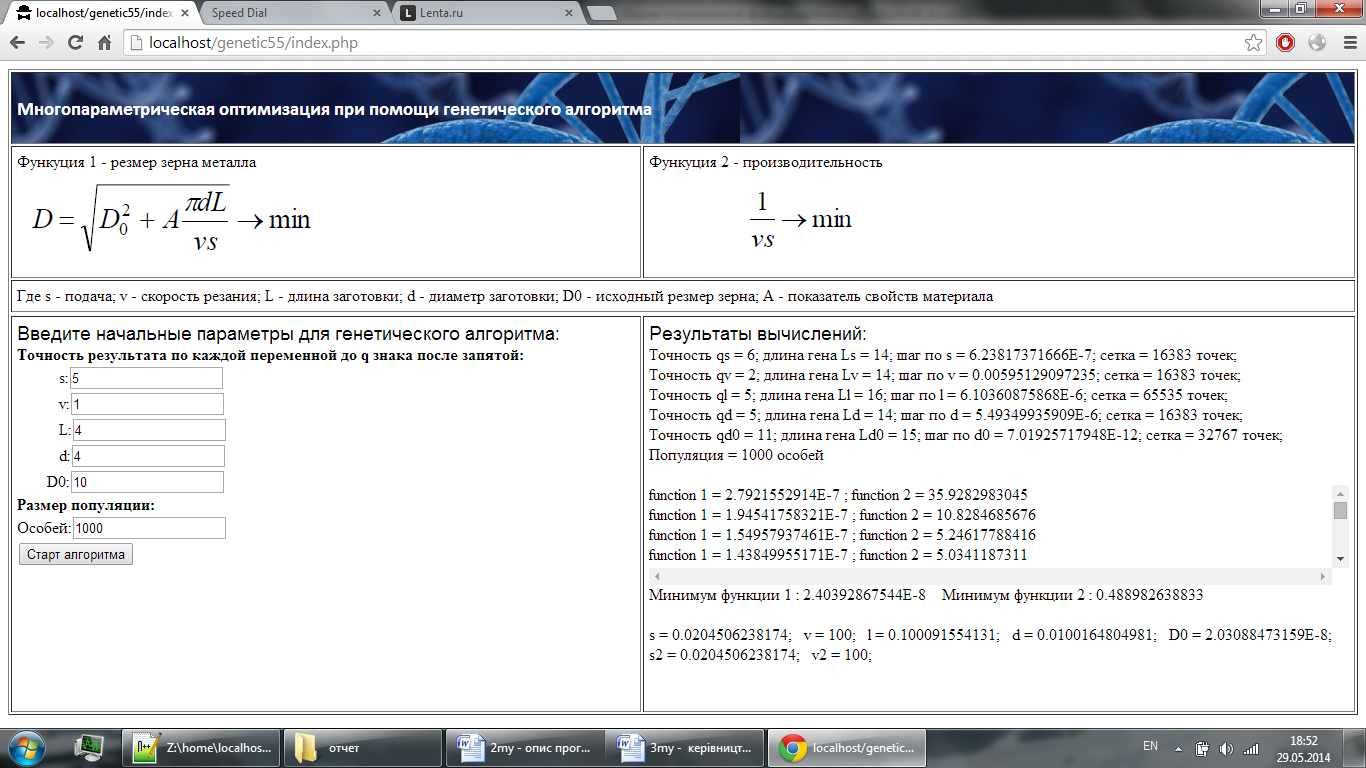


Рисунок 4.8 – розрахунок зі значеннями q = (6, 2, 5, 5, 11)

Результати розрахунків наведені у таблиці 4.2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| qs | qv | ql | qd | qd0 | F1 | F2 |
| 6 | 2 | 5 | 5 | 11 | 2.66604802797E-8 | 0.977517106549 |
| 6 | 2 | 5 | 5 | 10 | 2.67871149457E-8 | 0.977517106549 |
| 5 | 1 | 4 | 4 | 10 | 2.674485206E-8 | 0.977517106549 |
| 4 | 1 | 3 | 3 | 9 | 2.970077769E-8 | 0.988011077981 |
| 3 | 1 | 2 | 2 | 8 | 3.33417081704E-8 | 1.00431856985 |

Таблиця 4.2 – результати розрахунків

Проведені розрахунки показали, що зі значеннями qs = 5; qv = 1;

ql = 4; qd = 4; qd0 = 10; експерименти дають найбільш близькі один до одного значення цільових функцій. Використання менших коефіцієнтів дає більший розкид результатів, з тенденцією до погіршення. Використання більших значень не дає суттєвого покращення результатів, але значно підвищує час розрахунку, тобто їх використання не є раціональним рішенням.