Міністерство освіти і науки України

національний університет “Львівська політехніка”

****

Звіт

до розрахунково графічної роботи

На тему:

«jMetal фреймворк»

З курсу:

«Методи нечіткої логіки та еволюційні алгоритми при автоматизованому проектуванні»

Виконав:

ст. гр. СПКм-12

Баландюх О.А.

Перевірив:

Кривий Р.З

Львів 2016

**Мета роботи:**

Ознайомитись зі можливостями jMetal. Навчитись використовувати цей програмний продукт для вирішення певних задач.

**Огляд програмного продукту jMetal**

jMetal використовується для метаеврістічних алгоритмів в Java, це об'єктно-орієнтоване середовище на основі Java спрямоване на багатоцільовоу оптимізацію за допомогою метаеврістики. jMetal надає багатий набір класів які можуть бути використані в якості будівельних блоків багатоцільових методів; Таким чином, користуючись повторним використанням коду, jMetal надає можливість на одних й тих самих базових компонентах створювати алгоритми, такі як реалізація генетичних операторів і оцінок щільності, тим самим сприяючи не лише розвитку нових багатоцільовий техніки, а й проводити різні види експериментів. Наявність ряду класичних алгоритмів та алгоритмів, що використовують доволі “творчий” підхід до вирішення задачі надають новачкам не тільки можливість вивчити основні принципи багатоцільової оптимізації в метаеврістиці, а й дозволяють, без значних зусиль, застосувати їх для вирішення реальних проблем. Проект jMetal постійно розвивається. Так як він розробляється окремою групою розробників, а не компанією з виготовлення програмного забезпечення, нові версії випускаються тоді, коли розробники вважають, що потрібно додати новий функцій, які буде потрібним і актуальним для проведення дослідницької діяльності.

**Переваги jMetal**

До переваг фреймворку jMetal можна віднести наступне:

* Простота і зручність у використанні. Це ключові цілі: якщо вони не будуть виконані, мало хто буде використовувати програмне забезпечення. Класи, що надаються jMetal дотримується принципу, що кожен компонент повинен робити тільки одну річ, і робити це добре. Таким чином, базові класи (SolutionSet, Solution, Variable і т.д.) і їх операції інтуїтивно зрозумілі і, як наслідок, легко зрозуміти і використовувати. Крім того, каркас включає в себе реалізацію багатьох метаеврістичних класів, які можуть бути використані в якості шаблонів для розробки нових методів.
* Гнучкість. Це спільна мета. З одного боку, програмне забезпечення повинно включати простий механізм для виконання алгоритмів при різних налаштувань параметрів, включаючи специфічні параметри, а також ті параметри, які тісно пов'язані з проблемою, що потрібно вирішити. З іншого боку, такі питання, як вибір реального або двійкового представлення коду, відповідно до обраного типу представлення конкретні оператори для використання, повинні вимагати мінімальних змін в програмній реалізації.
* Переносимість. Фреймворк і алгоритми, розроблені з його використанням повинні виконуватись на машинах з різною архітектурою і / або бути готовими до запуску під різними операційними системами. Використання Java як мови програмування дозволяє з легкістю виконати це завдання; Крім того, дане програмне забезпечення не повинне бути повторно зкомпільоване для роботи в іншому середовищі.
* Можливість розширення. Нові алгоритми, оператори і вирішення проблем повинні бути легко додані в фреймворк. Ця мета досягається за рахунок використання деяких механізмів Java, таких як успадкування і пізнє зв'язування. Наприклад, всі MOPи успадковуються від класу Problem, тому нова проблема може бути створена написанням методу що зазначений в цьому класі; після того, як клас, який визначає нову проблему компілюється, більше нічого не повинно бути зроблено: механізм пізнього зв’язування дозволяє завантажувати код MOPа тільки тоді, коли цього потребує той чи інший алгоритм. Таким чином, jMetal дозволяє відокремити опис реалізації алгоритму від конкретної реалізації алгоритму.

**Інсталяція jMetal та виконання тестового алгоритму**

Фреймворк jMetal написано на Java, це значить, що він не вимагає будь-якого додаткового програмного забезпечення. Все що необхідно, це мати інстальованим Java JDK 1.5 або більш пізню версію. Вихідний код поставляється в пакеті tar.gz, який можна завантажити з [jmetal.sourceforge.net](http://jmetal.sourceforge.net).

Нижче буде представлено спосіб інсталяції jMetal використовуючи Eclipse IDE.

1. Розпакування вихідного коду фреймворка

Для подальшого використання фреймворку, нам необхідно видобути з архіву jmetal4.5.2.src.tar.gz весь його вміст в папку з назвою JMETALHOME.

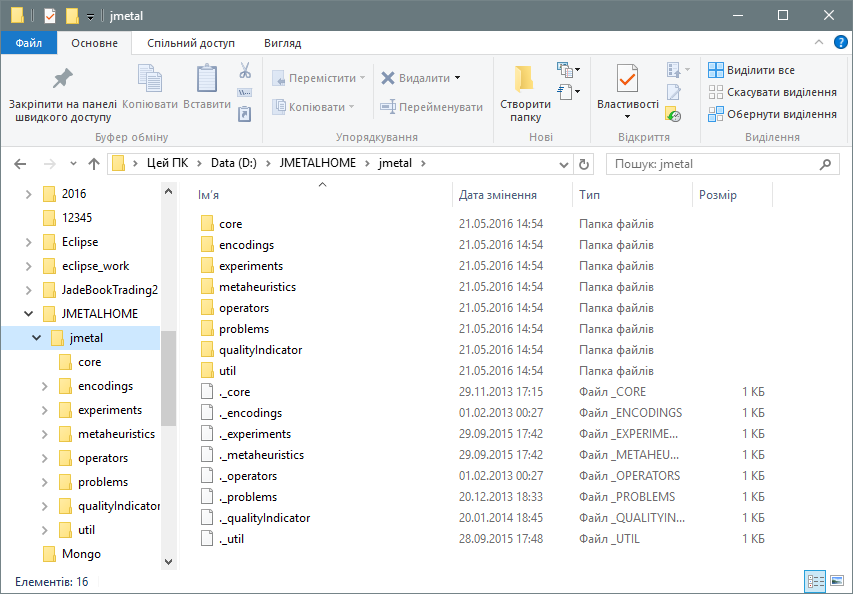


Рис. 1. Структура файлів вихідного коду jMetal

1. Підключення jMetal в Eclipse IDE.

Створення проекту:

* Виберіть *File* → *Create* → *New Java Project*.
* Введіть ім'я проекту (наприклад, jMetal) і натисніть на кнопку *Next*.
* Виберіть *Link additional source* і *Browse* щоб вказати розташування папки JMETALHOME (Рис. 2-3).
* Натисніть кнопку *Finish*.

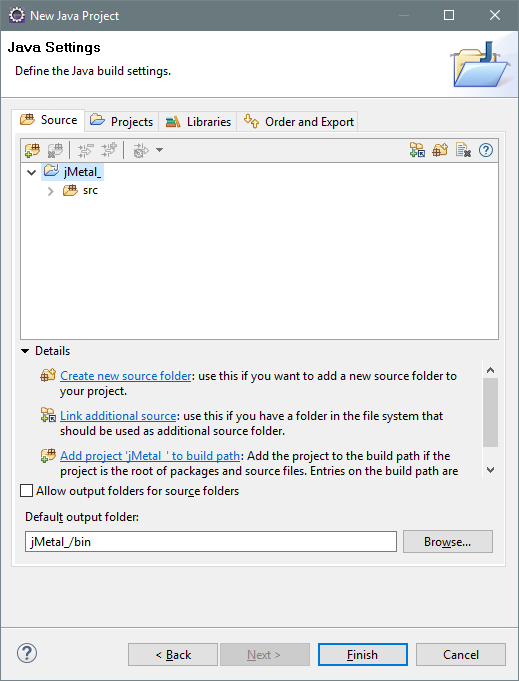


Рис.2. Вікно *New Java Project*

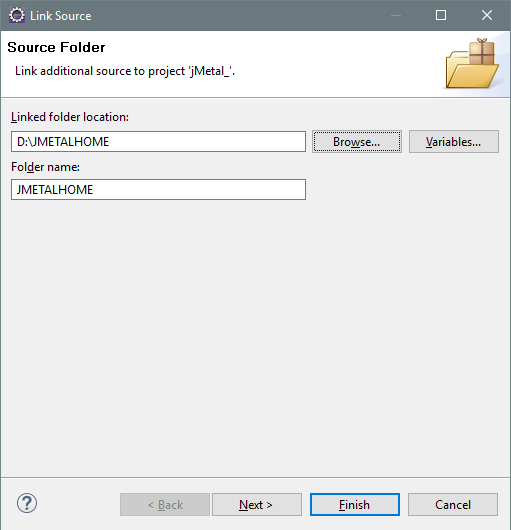


Рис.3. Вікно *Link Source*

1. Налагодження та виконання тестового алгоритму NSGA-II.

В останні роки було запропоновано низку багатокритеріальних еволюційних алгоритмів. Головною причиною цього є їх здатність знаходити декілька оптимальних за Парето розв’язків за один прохід. Розглянемо приклад реалізації багатокритеріального еволюційного алгоритма на основі недомінантного сортування (Non-dominated Sorting GA-II або NSGA-II). Завдяки своїм незначним вимогам до обчислювальних ресурсів та використанню елітизму, і реалізації підходу розділення придатності без використання спеціального параметра, алгоритм NSGA-II знаходить широке застосування.

Ми використовуємо NSGA-II (Non-dominated Sorting Genetic Algorithm-II) в якості прикладу. Алгоритм обраної задачі приведений на Рис.4.

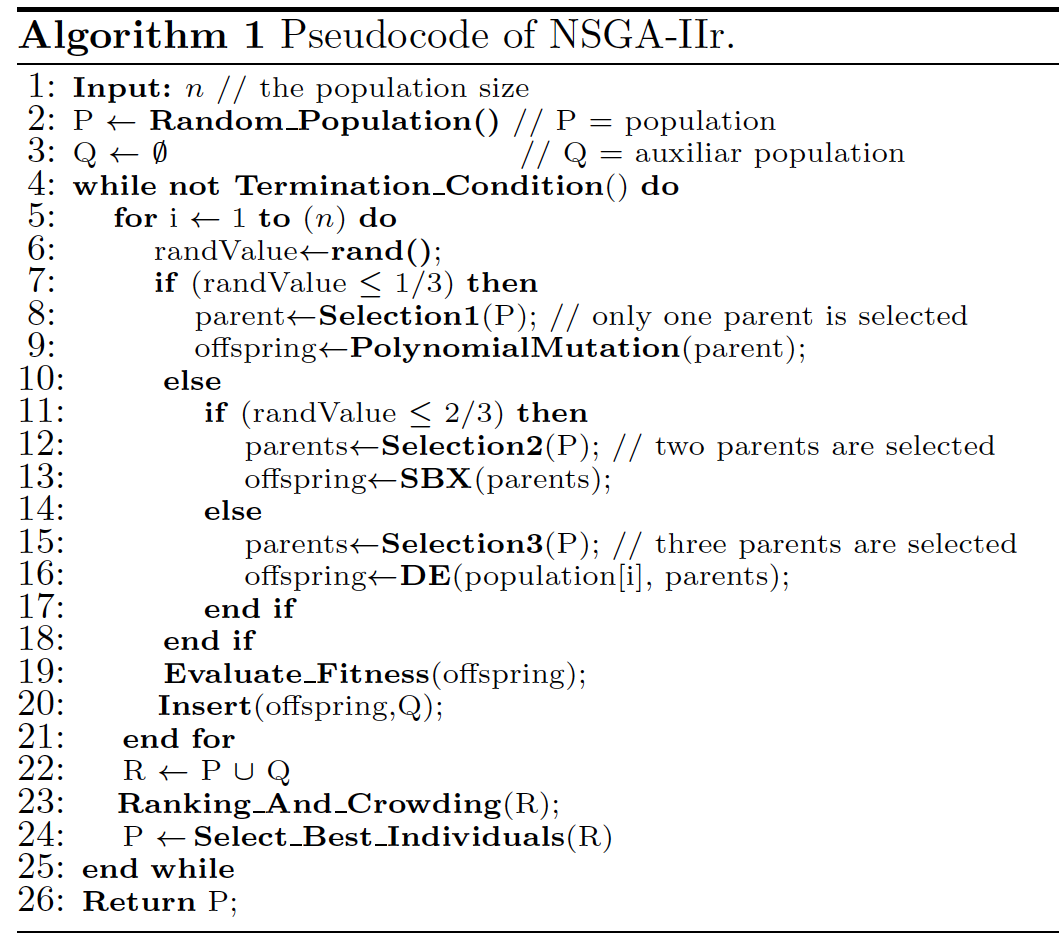


Рис.4. Алгоритм програми-прикладу

З наведеного вище алгоритму видно що дана програма є дуже гнучкою і легко піддається модернізації в залежності від поставлених вхідних даних і кількості параметрів. Дана програма має наступний набір вхідних параметрів:

* Problem – проблема для вирішення
* Algorithm – алгоритм що буде застусовуватися
* Crossover – кроссовер
* Mutation – вид мутації
* Selection – вид селекції

Мною було вибрані наступні значення вхідних параметрів генетичного

алгоритму:

* Problem – ZDT3
* Algorithm – NSGAII
* Crossover – SBXCrossover
* Mutation – PolynomialMutation
* Selection – BinaryTournament2

А також populationSize яке рівне 10, та maxEvaluations становить 2500.

Щоб налаштувати алгоритм, відкрийте файл NSGAII Main.java

вибравши його з пакета jmetal.metaheuristics.nsgaII (Рис.5). Після цього ви можете внести будь які зміни, чи удосконалення, в залежності від поставленої задачі. Код програмної реалізації NSGA-II знаходиться в додатку А.

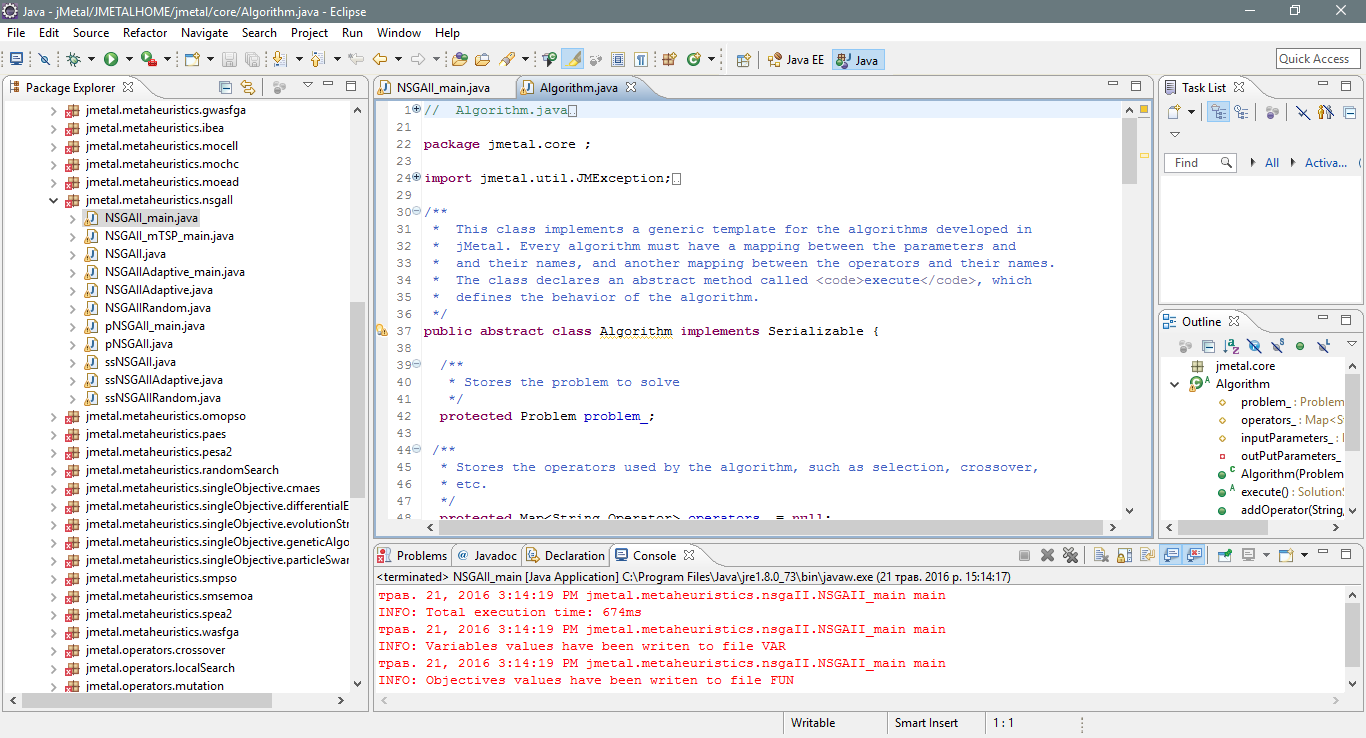


Рис.5. Представлення реалізації алгоритму NSGA-II в jMetal

Для запуску алгоритму, клацніть правою кнопкою миші на NSGA-II Main.java в дереві проекту. Виберіть *Run As* → *Java Application*. В результаті ви отримаєте два файли один з яких містить Парето-оптимальні рішення, а інший - фронт Парето, знайдені алгоритмом. За замовчуванням, ці файли називаються VAR і FUN, відповідно. Вони розташовані в каталозі *workspace* (наприклад, C:\Users\Oleg\workspace\jMetal). Вигляд *workspace* з файлами – результатом виконання програми зображенні на Рис.6. Результати виконання програми представлені в Додатку Б.

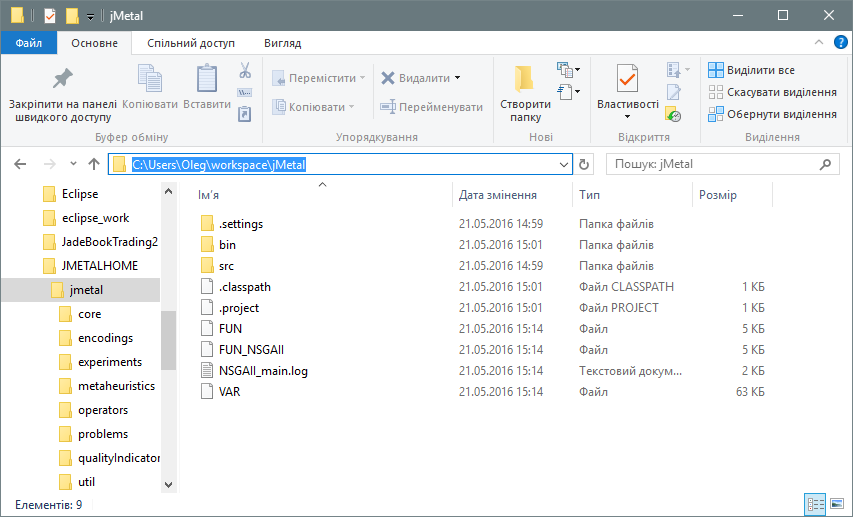


Рис.5. Вигляд *workspace* з файлами – результатом виконання алгоритму

**Висновок:** під час виконання розрахунково-графічної роботи, я ознайомилась з програмним продуктом jMetal, якй є надзвичайно зручним та ефективним інструментом для розв’язання різного роду задач, використовуючи евристичні та генетичні алгоритми. Фреймворк містить велику кількість вже реалізованих алгоритмів, а також набір задач з стандартними налаштуваннями.

Перевага використання цієї програми перш за все в тому, що для розв’язання задач немає необхідності реалізовувати певний алгоритм самостійно, а на томість можна використати чи модифікувати уже реалізований. Для цього фреймворк надає відкритий код своїх реалізацій алгоритмів.

Додаток А.

// NSGAII\_main.java

//

// Author:

// Antonio J. Nebro <antonio@lcc.uma.es>

// Juan J. Durillo <durillo@lcc.uma.es>

//

// Copyright (c) 2011 Antonio J. Nebro, Juan J. Durillo

//

// This program is free software: you can redistribute it and/or modify

// it under the terms of the GNU Lesser General Public License as published by

// the Free Software Foundation, either version 3 of the License, or

// (at your option) any later version.

//

// This program is distributed in the hope that it will be useful,

// but WITHOUT ANY WARRANTY; without even the implied warranty of

// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

// GNU Lesser General Public License for more details.

//

// You should have received a copy of the GNU Lesser General Public License

// along with this program. If not, see <http://www.gnu.org/licenses/>.

package jmetal.metaheuristics.nsgaII;

import jmetal.core.Algorithm;

import jmetal.core.Operator;

import jmetal.core.Problem;

import jmetal.core.SolutionSet;

import jmetal.operators.crossover.CrossoverFactory;

import jmetal.operators.mutation.MutationFactory;

import jmetal.operators.selection.SelectionFactory;

import jmetal.problems.ProblemFactory;

import jmetal.problems.ZDT.ZDT3;

import jmetal.qualityIndicator.QualityIndicator;

import jmetal.util.Configuration;

import jmetal.util.JMException;

import java.io.IOException;

import java.util.HashMap;

import java.util.logging.FileHandler;

import java.util.logging.Logger;

/\*\*

\* Class to configure and execute the NSGA-II algorithm.

\*

\* Besides the classic NSGA-II, a steady-state version (ssNSGAII) is also

\* included (See: J.J. Durillo, A.J. Nebro, F. Luna and E. Alba

\* "On the Effect of the Steady-State Selection Scheme in

\* Multi-Objective Genetic Algorithms"

\* 5th International Conference, EMO 2009, pp: 183-197.

\* April 2009)

\*/

public class NSGAII\_main {

public static Logger logger\_ ; // Logger object

public static FileHandler fileHandler\_ ; // FileHandler object

/\*\*

\* @param args Command line arguments.

\* @throws JMException

\* @throws IOException

\* @throws SecurityException

\* Usage: three options

\* - jmetal.metaheuristics.nsgaII.NSGAII\_main

\* - jmetal.metaheuristics.nsgaII.NSGAII\_main problemName

\* - jmetal.metaheuristics.nsgaII.NSGAII\_main problemName paretoFrontFile

\*/

public static void main(String [] args) throws

JMException,

SecurityException,

IOException,

ClassNotFoundException {

Problem problem ; // The problem to solve

Algorithm algorithm ; // The algorithm to use

Operator crossover ; // Crossover operator

Operator mutation ; // Mutation operator

Operator selection ; // Selection operator

HashMap parameters ; // Operator parameters

QualityIndicator indicators ; // Object to get quality indicators

// Logger object and file to store log messages

logger\_ = Configuration.logger\_ ;

fileHandler\_ = new FileHandler("NSGAII\_main.log");

logger\_.addHandler(fileHandler\_) ;

indicators = null ;

if (args.length == 1) {

Object [] params = {"Real"};

problem = (new ProblemFactory()).getProblem(args[0],params);

} // if

else if (args.length == 2) {

Object [] params = {"Real"};

problem = (new ProblemFactory()).getProblem(args[0],params);

indicators = new QualityIndicator(problem, args[1]) ;

} // if

else { // Default problem

//problem = new Kursawe("Real", 3);

//problem = new Kursawe("BinaryReal", 3);

//problem = new Water("Real");

problem = new ZDT3("ArrayReal", 30);

//problem = new ConstrEx("Real");

//problem = new DTLZ1("Real");

//problem = new OKA2("Real") ;

} // else

algorithm = new NSGAII(problem);

//algorithm = new ssNSGAII(problem);

// Algorithm parameters

algorithm.setInputParameter("populationSize",10);

algorithm.setInputParameter("maxEvaluations",2500);

// Mutation and Crossover for Real codification

parameters = new HashMap() ;

parameters.put("probability", 0.9) ;

parameters.put("distributionIndex", 20.0) ;

crossover = CrossoverFactory.getCrossoverOperator("SBXCrossover", parameters);

parameters = new HashMap() ;

parameters.put("probability", 1.0/problem.getNumberOfVariables()) ;

parameters.put("distributionIndex", 20.0) ;

mutation = MutationFactory.getMutationOperator("PolynomialMutation", parameters);

// Selection Operator

parameters = null ;

selection = SelectionFactory.getSelectionOperator("BinaryTournament2", parameters) ;

// Add the operators to the algorithm

algorithm.addOperator("crossover",crossover);

algorithm.addOperator("mutation",mutation);

algorithm.addOperator("selection",selection);

// Add the indicator object to the algorithm

algorithm.setInputParameter("indicators", indicators) ;

// Execute the Algorithm

long initTime = System.currentTimeMillis();

SolutionSet population = algorithm.execute();

long estimatedTime = System.currentTimeMillis() - initTime;

// Result messages

logger\_.info("Total execution time: "+estimatedTime + "ms");

logger\_.info("Variables values have been writen to file VAR");

population.printVariablesToFile("VAR");

logger\_.info("Objectives values have been writen to file FUN");

population.printObjectivesToFile("FUN");

if (indicators != null) {

logger\_.info("Quality indicators") ;

logger\_.info("Hypervolume: " + indicators.getHypervolume(population)) ;

logger\_.info("GD : " + indicators.getGD(population)) ;

logger\_.info("IGD : " + indicators.getIGD(population)) ;

logger\_.info("Spread : " + indicators.getSpread(population)) ;

logger\_.info("Epsilon : " + indicators.getEpsilon(population)) ;

int evaluations = ((Integer)algorithm.getOutputParameter("evaluations")).intValue();

logger\_.info("Speed : " + evaluations + " evaluations") ;

} // if

} //main

} // NSGAII\_main

Додаток Б.

Вміст файлу FUN

0.8527480816882838 -0.531346821232097

2.0462432970769218E-5 1.4053434185786629

0.618408024892632 0.16461963008522293

0.39923957727516 0.6627624861890485

0.41760487243752975 0.43717417659603136

0.19422892894411575 0.9592618928072758

0.1815280456467831 1.0207125497004856

0.010471671794589188 1.3049253367656943

0.6304694058489057 -0.024203690448263777

0.8160366746332643 -0.03878415262140786

Вміст файлу FUN\_NSGAII

0.8527480816882838 -0.531346821232097

2.0462432970769218E-5 1.4053434185786629

0.618408024892632 0.16461963008522293

0.39923957727516 0.6627624861890485

0.41760487243752975 0.43717417659603136

0.19422892894411575 0.9592618928072758

0.1815280456467831 1.0207125497004856

0.010471671794589188 1.3049253367656943

0.6304694058489057 -0.024203690448263777

0.8160366746332643 -0.03878415262140786

Вміст файлу VAR

0.8527480816882838 0.0031850562410707115 0.013896898501979092 0.010468008780772954 0.03605536253724006 1.171752252857888E-4 0.0011975635924326468 0.0301695095837236 0.26778852601451936 1.784173262546684E-4 0.007552134889691393 0.01719122533224983 0.011868932013768705 0.18435944632699638 0.005848879124768847 0.008796698383412675 0.0013669858775593976 0.008351219323289659 0.15596167197005817 0.0635287468587868 0.0016955047495447514 0.024994248809975066 0.04113308028816665 0.0041553324272369475 0.019653443770553513 0.01101989179358654 0.011402127294081073 5.775421394258011E-4 0.4035648171680505 7.41690264673291E-5

2.0462432970769218E-5 0.003195089664489382 0.013896084119061135 0.02027478674567046 0.009064990924506014 1.1958353809054003E-4 0.016089276416371052 0.029577877737677366 0.24712008809920666 8.142274570251616E-4 0.006800323951216025 0.01703134027208643 0.012947203438674693 0.18430828974830257 0.007398444410288565 5.142434214013124E-4 0.0021846426426926817 0.00434423734353151 0.15978229720577386 0.062053709850590506 0.0048130879375570135 3.6837792982981864E-4 0.04118918528560649 0.004683248766560563 0.017789108416733413 0.011032316133492317 0.01146170110251348 7.937661917558187E-4 0.43274873458812796 0.0010226173976212955

0.618408024892632 0.0031833534851705278 0.012806027628354514 0.020087490261661584 0.027492314602514743 2.423102135044582E-4 0.016482537561107152 0.029907449499259282 0.30691574940297234 0.001046713101435356 0.007552113524599852 0.017037846033375433 0.012968613344455279 0.1845075665185047 0.004807298223160052 0.008768388793037457 8.262797852586238E-5 0.008388846108663127 0.1577222607753242 0.101874058911226 0.011339481614977113 0.025087709075774768 0.041137065699906286 0.004312607123230665 0.01752991237904309 0.010035793588131146 0.014417886709959273 0.001117436469103117 0.40102664507757996 1.260589992158596E-4

0.39923957727516 0.003195089664489382 0.013896084119061135 0.02027478674567046 0.009157253724895799 1.1958353809054003E-4 0.016089276416371052 0.029696543010137123 0.24905262224680166 8.142274570251616E-4 0.006800323951216025 0.01703388695867095 0.008860386738159887 0.1843129978448787 0.007398444410288565 5.436664689022345E-4 0.0021846426426926817 0.001796749110031269 0.15978229720577386 0.062053709850590506 0.004882435997286423 3.6837792982981864E-4 0.04118918528560649 8.553466447069459E-4 0.017789108416733413 0.011014512108911695 0.01146170110251348 7.875941890811951E-4 0.4101833301581082 0.0010118373801855135

0.41760487243752975 0.0029375873106295986 0.013686358672004566 0.010150935536671336 0.03605536515157146 0.0010156016515360609 0.016405716836851773 0.029624260643175932 0.2645876567473846 0.0010490298911048608 0.0071494160424026626 0.017029134372382564 0.011846840175221064 0.1843881015457034 0.048123330090159365 0.003917284989493118 0.002209550800034343 0.008441926029228537 0.1577026207019557 0.06324141728707303 0.008554157798907567 0.0021116077127325927 0.04119591844880611 0.004208476313068149 0.019485097628868187 0.010695274782571335 0.011512996894457775 7.977965213271888E-4 0.40361388445978946 7.937142069846057E-4

0.19422892894411575 0.0031850562410707115 0.013896898501979092 0.02019141229815998 0.04401345833767222 1.1700304296927015E-4 0.0011975635924326468 0.0301695095837236 0.3563046247227044 1.871656639650645E-4 0.007645138929956318 0.01703682182433113 0.012915182153614072 0.18435944632699638 0.005848879124768847 0.003527080257875606 0.0013482774424299112 0.008388859334858058 0.15361466093478968 0.07515956400083908 0.0016955047495447514 0.009748775845231119 0.04113308028816665 0.00468299144746289 0.017833316352487992 0.011024665532914816 0.011402127294081073 5.775421394258011E-4 0.4035648171680505 0.028670416111592892

0.1815280456467831 0.0031901707101812237 0.013896898501979092 0.02026961101877282 0.03609705404726443 1.1717886856589069E-4 0.0011975635924326468 0.029691325115581108 0.26769828579354016 1.7827371219870448E-4 0.004010867422498571 0.01703682178287998 0.012915185043170628 0.18450653401645664 0.0075648588247931485 0.009031543509607578 0.001445955629442303 0.028826437375280956 0.15367255648700145 0.0635287468587868 0.011357726853301475 0.02457900674108925 0.04113259651167606 0.004141415497013231 0.019587641759631853 0.01030537998585108 0.011402127294081073 5.775421394258011E-4 0.41016130471952833 9.378293217155782E-5

0.010471671794589188 0.0029346668350249245 0.021728957313945783 0.020190873644308632 0.008338436317914656 1.1975972021475603E-4 0.0012525545771234145 0.03311900214031188 0.30728335100919496 1.9458824516924253E-4 0.007546973470980181 0.01703740345565359 0.012929073311630221 0.18435163278651653 0.005828392523526597 0.008819329673541897 0.00221580923272592 0.008593336868384302 0.15588370047454017 0.06352167171943209 0.008504563029425191 0.02445348758461253 0.041132705225420736 0.00468299049244711 0.01775744563236042 0.010654312098436945 0.011402695231925155 0.0010389241603275465 0.4053010523574936 0.0010227886601900642

0.6304694058489057 2.638867638568151E-4 0.012734127812776781 0.020087490261661584 0.027492314602514743 2.423102135044582E-4 0.016482469013107848 0.029907449499259282 0.30691574940297234 0.001046713101435356 0.007550187778375164 0.017037846033375433 0.0129623364083268 0.18450688712178917 0.004754902471494031 0.008768388793037457 0.0014606967234301758 0.0077996154998908815 0.15771764885586448 0.101874058911226 0.011357784020902872 0.025087709075774768 0.04114091748969854 0.004463603574180458 0.01752991237904309 0.010312217573793088 0.006301911322592926 5.453681604073867E-4 0.40102664507757996 9.705125801594402E-5

0.8160366746332643 0.0029346668350249245 0.014029923765299471 0.01982997935386112 0.03605536253724006 1.1723888907844538E-4 0.0012525545771234145 0.029992217515006674 0.2686557461469433 1.875158330176398E-4 0.007552134889691393 0.017040728930262487 0.012947407227792284 0.18435163278651653 0.005849541177925333 0.008798370049318785 0.0013420008577004097 0.008356786917670384 0.15588370047454017 0.06352874678596133 0.008592116541213963 0.024646444621261566 0.04113308028816665 0.00468299144746289 0.01963408314372227 0.010655711070246529 0.011402695231925155 0.05043224582157062 0.403611653603539 6.949696755146766E-5