Non-dominated Sorting on Two Objectives

Michael Mazurek
Canadian Forces Aerospace Warfare Centre OR Team Coop Student

Slawomir Wesolkowski, Ph.D.
Canadian Forces Aerospace Warfare Centre OR Team

DRDC CORA TN 2009-027
July 2009

Defence R&D Canada
Centre for Operational Research and Analysis

National Defence Défense nationale

Canada

Non-dominated Sorting on Two Objectives

Michael Mazurek
Canadian Forces Aerospace Warfare Centre OR Team Coop Student

Slawomir Wesolkowski, Ph.D.
Canadian Forces Aerospace Warfare Centre OR Team

Defence R&D Canada – CORA

Technical Note
DRDC CORA TN 2009-027
July 2009

Principal Author

Original signed by Michael Mazurek *Slawomir Wesolkowski*

Michael Mazurek Slawomir Wesolkowski

Defence Research Assistant Defence Scientist

Approved by

Original signed by D.M. Bergeron, Ph.D.

D.M. Bergeron, Ph.D.

Head Air Section

The information contained herein has been derived and determined through best practice and adherence to the highest levels of ethical, scientific and engineering investigative principles. The reported results, their interpretation, and any opinions expressed therein, remain those of the author and do not represent, or otherwise reflect, any official opinion or position of DND or the Government of Canada.

This research was sponsored by the Canadian Forces Aerospace Warfare Centre (CFAWC) as part of Project Yukon.

Defence R&D Canada – Centre for Operational Research and Analysis (CORA)

Abstract

The fast and elitist non-dominated sorting genetic algorithm (NSGA-II) contains a mechanism to sort individuals in a multi-objective optimization problem into non-dominated fronts, based on their performance in each optimization variable. When dealing with a bi-objective problem it is possible to carry-out the non-dominated sorting more efficiently, using the new sorting method presented in this paper. Results comparing the efficiency of the two sorting mechanisms on sets of data with two objectives are presented and discussed.

This page intentionally left blank.

Table of contents

List of figures

List of tables

Acknowledgements

The authors thank Dr. Andrew Billyard and Ms. Irene Collin for their helpful comments on how to improve this technical note.

1 Introduction

Many real world problems such as fleet mix optimization [1], [2], [3] have usually two or more conflicting objectives which cannot be adequately analyzed using single-objective optimization algorithms. Therefore, we refer to them as multi-objective problems. For example, in a fleet-mix problem platform cost would conflict with mission duration (assuming that the more expensive platforms are the faster ones and vice versa). In general, a multi-objective problem can be characterized by:

$$\begin{array}{lll} \min\limits_{\vec{x}} & f_m(\vec{x}) & m = 1, \ldots, M \\ \text{s.t.} & g_j(\vec{x}) \le 0 & j = 1, \ldots, J \end{array} \qquad (1)$$

where $f_m(\vec{x})$ represents the *m*-th of *M* objective functions and $g_j(\vec{x})$ the *j*-th constraint of a set of *J*. (Note that any objective function for which a maximized objective score is desirable can be transformed into a minimization problem simply by multiplying the function by -1. This paper assumes that all optimization problems are minimization problems, without loss of generality.).

Since conflicting objectives $f_m(\vec{x})$ require trade-offs, they will most likely result in solutions which are not optimal in any one of the objectives. If all solutions in this non-optimal set are not dominated by each other they form the non-dominated solution set. (A solution A is said to dominate solution B if at least one of the objectives for solution A is less than that of solution B, and all other objectives are either less than or equal to each other). This non-dominated solution set is called the Pareto front [4]. All the solutions on the Pareto front are equally valid as no one solution is superior to any other over all objectives. The decision maker can then use this set to decide on the relative importance of each objective.

In this paper, we propose to improve the non-dominated sorting mechanism in the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [5]. Note that we only improve NSGA-II in computational complexity allowing it to handle much larger population sets (i.e., other parts of the algorithm still function in the same way as before). In Section 2 we will describe NSGA-II [5], paying particular attention to the sorting algorithm. NSGA-II has been widely used since its introduction in 2002 and is well known. The total worst-case complexity of NSGA-II (when sorting a population of size *N*) using its original sorting method is $O(M \cdot N^2)$ for either *1* or *N* non-dominated fronts (i.e. either every member of the population is in the same non-dominated front, or each member is in a separate front).

In Section 3, we describe our improved non-dominated sorting algorithm. For a bi-objective problem (i.e. for *M=2* objectives), the improved algorithm takes advantage of preliminarily sorting the first objective and then analyzing the individuals in order, along the first objective. Since the second objective conflicts with the first, there is no need to explicitly analyze it. The best and worst case complexities are (respectively) $O(N \log N)$ if the whole population of *N* solutions is in one front and $O(N^2)$ in the case where there are *N* fronts. Naturally, the worst-case computational complexity of NSGA-II's sorting method reduces to $O(N^2)$ with *M=2* objectives. At this time, the improved non-dominated sorting algorithm can only be used on problems with two objectives. We give a counter example of why it cannot be used with three or more objectives in Section 4.

In Section 5, we present computational complexity results showing that the new non-dominated sorting algorithm is indeed several orders of magnitude faster (the larger the population the larger the increase in speed) than the algorithm used in NSGA-II (thanks primarily to the logarithmic complexity gains). We present results on the number of performed comparisons. We show actual running times in a Matlab® implementation for illustrative purposes, and the new method performs 2-3 times faster than the NSGA-II sorting mechanism. This increase in speed is important, as it will allow the genetic algorithm to be run with larger population sizes, without a significant cost in running time. The paper concludes with Section 6.

2 NSGA-II

NSGA-II is an elitist evolutionary algorithm which groups individual solutions into non-dominated fronts, and also contains a crowding-distance assignment operator to preserve diversity in the population of solutions **[5]**.

2.1 Description of the Algorithm

For a population of size *N*, a group of *N* offspring are created using the desired crossover and mutation operators. To ensure elitism, the offspring and their parent solutions are combined into a group of size *2N*, and the fittest *N* individuals from this group are used as parents for the next generation. To determine the top *N* fittest individuals, the solutions are assigned to non-dominated fronts. The first front is found by comparing all individuals in the population (in the objective space), and assigning the individuals which are not dominated by any other to the first front. This front is set aside and removed from the current population, at which point the non-dominated front from the remaining population members is found, which then becomes the second front. Each new front is removed from the population and set aside, until all fronts have been found.

The fronts are then combined (in order) to create the parents for the next generation. It is unlikely that the final front to be included will make exactly *N* parents, so an operator is needed to truncate the front. To preserve diversity in the final group of solutions, NSGA-II removes the “most crowded” individuals from this final front, in order to make it fit into the group of *N* parents. To determine which individuals are the most crowded, each individual in the front is assigned a crowding-distance. An individual’s crowding-distance is defined as the sum (over all objectives) of the distance between its two closest neighbours in each objective. If an individual is at one of the endpoints for an objective (i.e. scores either the maximum or minimum value in the objective’s fitness function), then it is assigned a crowding-distance of infinity. The first individuals to be removed are ones with the smallest crowding-distance.

After the offspring and parent individuals have been combined, sorted into fronts, and recombined into a new parent population (after truncating a front based on crowding-distance, if necessary), a new group of offspring is created, and the algorithm runs for a set number of generations, or until some other stopping criterion is reached. The initial parent population is selected randomly from the solution space.

Fig. 1 provides an illustration of one generation of NSGA-II. In Step 1, the parent and child populations are combined. Step 2 sorts the combined populations into non-dominated fronts, and if necessary, a front is truncated in Step 3. The top *N* individuals are the parents for the next generation.

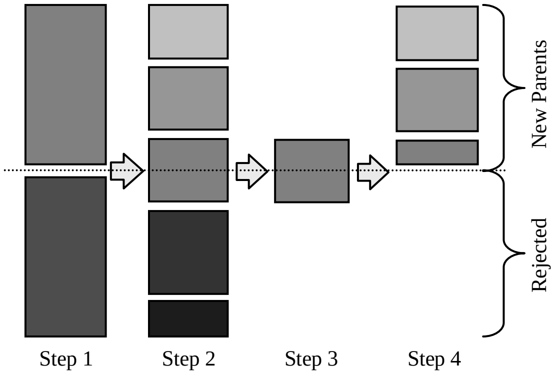

Figure 1: Illustration of NSGA-II generation

2.2 NSGA-II Sorting Technique

To determine which individuals from a population belong to the non-dominated front (NDF), NSGA-II does a comparison between all members of the population and a dynamically changing non-dominated set (as illustrated in Fig. 2). The main idea is that a non-dominated set (NDS) of solutions is maintained, each population member is compared with every member of this set at some point, and the members of the NDS are updated after every comparison.

Firstly, any two solutions in the population, A and B, are compared to each other. There are three possible results from this comparison:

1. A dominates B
2. B dominates A
3. A and B are incomparable (i.e. neither dominates the other)

If A and B are incomparable, they become the first two members of the NDS. If A dominates B, then A becomes the first member of the NDS, and B is set aside. Similarly, if solution B dominates A, then A is set aside, and B becomes the first member of the NDS. Each of the remaining population members are then compared, in order, with each NDS member. Any NDS members which are dominated by a population member are removed from the NDS. If a population member is not dominated by any NDS member, it is added to the NDS, otherwise, it is set aside, as it is not part of the NDF. After searching through all individuals in the population in this way, the final members in the non-dominated set will constitute the non-dominated front of the population.

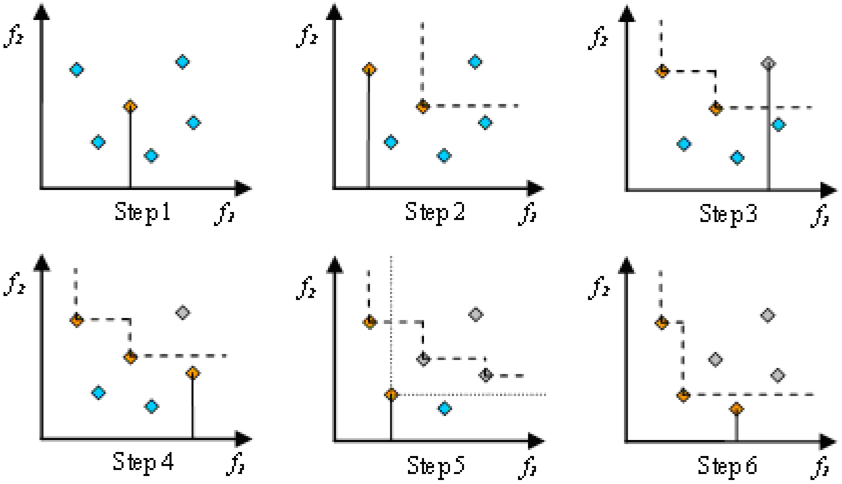

Figure 2: Finding a non-dominated front with the NSGA-II technique

2.3 Computational Complexity

There are two worst-case scenarios for this sorting technique. If every individual in a population of size *N* is in the same non-dominated front, then the second individual will be compared to the first one, the third will be compared to the first two, and the *N*-th individual will be compared to the first *N*-1 individuals, for a total of *N(N-1)/2* comparisons, which corresponds to a computational complexity of $O(N^2)$ for the non-dominated sort. On the other end of the spectrum, if every individual is in its own front, then it will take *N-1* comparisons to find the first front, *N-2* comparisons to find the second front, one comparison to find the (*N-1*)-th front, and no comparisons to find the *N*-th front, for a total complexity of $O(N^2)$. Furthermore, since each comparison between two individuals requires *M* objective function value comparisons (if *M* is the number of objectives), the total worst-case complexity of this sorting method is $O(M \cdot N^2)$ for both cases of either *1* front or *N* separate fronts.

3 Improved Non-dominated Sorting on Two Objectives

The proposed sorting technique is similar to the method used in NSGA-II, in that to sort a population into non-dominated fronts, the first non-dominated front is found and removed from the population, then the second is found from the remaining members and removed, then the third, and so on, until every member of the population has been assigned to its proper front. However, the method by which the non-dominated fronts are found (illustrated in Fig. 3) differs significantly between the two methods.

3.1 Description of Improved Technique

For a population of individuals with M=2 objectives, the individual with the lowest first-objective score must be part of the NDF, since it cannot be dominated by any other individual. (If two or more individuals tie for the lowest first-objective score, then these solutions must also be compared in the second objective, and the individual(s) which score(s) best in the second objective are definitely in the population's first non-dominated front. The proposed algorithm repeatedly applies this idea to efficiently find the population's NDF.

As a first step, the individuals are ordered lexicographically, primarily by their first-objective scores, and then by their second-objective scores to break any ties. Clearly, sorting the individuals in this way ensures that no individual can be dominated by any which come after it in the order, and hence, the first individual in this sorted order must be a member of the NDF. The individuals are then iterated through in order, and the first one not dominated by the first NDF member must also be part of the NDF, where it becomes the *most-recent NDF member*, and the first NDF member is known as the *previous* NDF member.

To determine the non-dominance of successive individuals, they do not need to be compared to both NDF members, but rather only the most-recent one. Since the most-recent NDF member is incomparable with the previous NDF member, and also since the previous member has a lower first-objective score than the most-recent member, the most-recent NDF member's second-objective score must be lower than the previous member's. Additionally, since all remaining individuals have higher first objective scores than both the previous and most-recent NDF members, any individuals that are incomparable with the most-recent member must also be incomparable with the previous NDF member. If an individual is added to the NDF, the most-recent NDF member then becomes the previous member, and the newly added individual is the most-recent member.

By initially sorting the individuals by their first objective scores, only one comparison is needed to determine whether or not an individual is added to the front. Each individual is either dominated by the most recent NDF member, and therefore is not part of the non-dominated front, or it is incomparable with the NDF member, in which case it is incomparable with all NDF members, and can be added to the non-dominated front.

An example of sorting a dataset on two objectives is illustrated in Figure 3. The key difference between this proposed algorithm and the method used in NSGA-II is that NSGA-II's method has to compare individuals to every member of the dynamically changing non-dominated set of individuals, while the proposed algorithm can use the information gained from sorting the population by their first-objective scores to intelligently reduce the number of comparisons required between individuals.

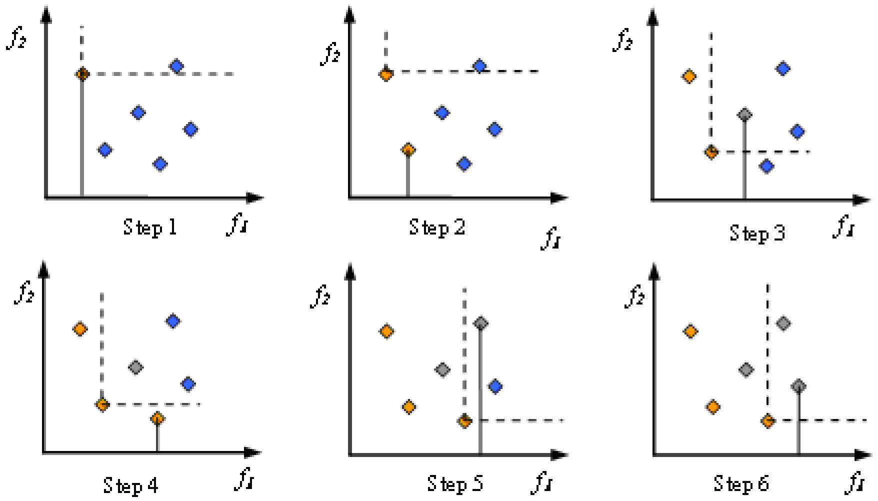

Figure 3: Finding the non-dominated front with the proposed algorithm

3.2 Computational Complexity

To find the first non-dominated front from a population of N individuals, using the merge sort algorithm [6], the initial sorting has a worst-case computational complexity of $O(M \cdot N \log N)$, for $M=2$ objectives, (when every individual has the same score in the first objective, and therefore has to be sorted over both objectives). The initial sorting allows the first non-dominated individual to be found, but then only $N-1$ comparisons need to be performed to determine the remaining members of the non-dominated front, since either counting or discounting an individual in the front takes only one comparison. Therefore the overall complexity of finding the first non-dominated front is $O(N \log N)$. To find subsequent fronts, the initial sorting does not have to be repeated, as the members of the non-dominated front can just be removed from the sorted list (which takes k steps, where k is the number of individuals in the front), and the next front can be found with only $N-k-1$ steps.

In the worst case for the proposed sorting algorithm, each individual will be in its own front (i.e. there will be a total of N fronts). The initial sort will have a complexity of $O(N \log N)$, and $N-1$ comparisons are needed to find the first front, $N-2$ are needed for the second, $N-3$ for the third, and so on, for a total complexity of $O(N^2)$. This is no worse than NSGA-II's complexity, in this case. However, in NSGA-II's other worst-case scenario, when every population member is part of the first non-dominated front, the proposed algorithm performs much better. In fact, this is the proposed algorithm's best-case scenario. The initial sorting still has a complexity of $O(N \log N)$, but only $N-1$ comparisons are needed to determine that every other individual in the population belongs to the non-dominated front, which gives this scenario an overall complexity of $O(N \log N)$. Empirical results from a comparison of the proposed technique with NSGA-II's sorting method in less extreme cases (i.e. populations with between 1 and N non-dominated fronts) are discussed in Section V.

4 Sorting on Three or More Objectives

Although the proposed sorting technique performs quite well when it is sorting based on only two objectives, it breaks down when a problem with three or more objectives is considered. When more than two objectives are considered, the proposed algorithm is not discriminating enough when choosing individuals to be included in the non-dominated front, and it will return a front with too many individuals. Although all individuals that are part of the first non-dominated front are included in the first generated front, to remove those extra individuals or solutions increases the computational complexity to the point where the proposed algorithm becomes just as expensive as NSGA-II's sorting method.

4.1 A Simple Counter-Example

Consider the sorting of the three individuals shown in Table 1. Each individual is sorted over three objectives, and they are initially sorted by how they score in the first objective. Assume that a lower objective score corresponds to a more optimal one (i.e. this is a minimization problem).

If the proposed algorithm was used to sort these three individuals, as a first step, Individual 1 would be added to the non-dominated front. Next, Individual 2 would be compared with Individual 1, and, since these two individuals are incomparable, Individual 2 would be added to the non-dominated front. Finally, as a third step, Individual 3 would be compared to Individual 2, and then added to the non-dominated front, since Individuals 2 and 3 are incomparable. However, Individual 3 is dominated by Individual 1, so it does not belong in the non-dominated front. Since the proposed technique is not guaranteed to find the correct non-dominated front in cases where populations with three or more objectives need to be sorted, the proposed sorting technique should only be used when dealing with a problem with two objectives.

Table 1: Example Three-Objective Population

Individual	**Objective Function Scores**		
	Objective 1	**Objective 2**	**Objective 3**
1	0	0	0
2	1	4	-1
3	2	2	2

5 Results

Both the proposed sorting technique and the sorting method used in NSGA-II were implemented in Matlab® on a 3.20 GHz Intel® Pentium® 4 processor, with 1 GB available memory, on randomly generated sets of data.

5.1 Total Number of Comparisons

Tests were run for six different population sizes, and ten random populations were generated for each of these population sizes. Each of these populations was sorted using both the proposed method and NSGA-II's method. Of course, the sorting was done based on two objectives.

Table 2: Number of Comparisons for Both Sorting Methods

Population Size	**Proposed**		**NSGA-II**	
	Average (x10^3)	**Std Dev (%)**	**Average (x10^3)**	**Std Dev (%)**
100	0.7	7.47	1.7	5.55
200	2.2	8.43	5.8	7.29
400	6.3	4.88	18.0	2.80
800	18.1	4.55	55.1	5.15
1600	52.4	3.90	170.2	3.19
3200	148.4	2.39	519.5	2.11

Table 2 shows both the average and standard deviation of the number of comparisons between individuals (over the ten tests for each population size) for both sorting methods. Note that the overhead $O(M \cdot N \log N)$ sorting complexity in the proposed method is not included in this table, only the number of direct comparisons between individuals is presented.

The results in Table 2 show that in the expected case, the proposed method is able to sort a population of individuals (with two objectives) with significantly fewer steps than the method used in NSGA-II. Even if the overhead $O(M \cdot N \log N)$ sorting complexity is added to the average number of comparisons counted in the proposed method, the proposed method still performs notably fewer comparisons than the NSGA-II mechanism.

5.2 Running Time

Table 3 shows the average and standard deviation of the running times for the same tests as are presented in Table 2. For small populations (less than 800), the proposed algorithm runs about twice as fast as NSGA-II's sorting technique in this implementation, while for larger populations, the proposed method is almost three times quicker. Clearly, the proposed technique is a much faster alternative to NSGA-II's sorting method.

Table 3: Running Times for Both Sorting Methods

Population Size	Proposed		NSGA-II	
	Average (ms)	Std Dev (%)	Average (ms)	Std Dev (%)
100	43	14.3	68	9.13
200	121	9.30	226	11.9
400	321	5.11	672	4.09
800	881	5.22	2030	5.64
1600	2538	4.46	6395	3.45
3200	7117	2.03	20147	1.61

6 Conclusion

A non-dominated sorting technique was proposed for populations which are evaluated in two objectives, intended to replace the current sorting method used by NSGA-II. Although the proposed method's computational complexity has the same worst-case scenario as NSGA-II's technique, it improves significantly on one of NSGA-II's two worst-case scenarios, and it was shown empirically to outperform the NSGA-II method in a typical sorting situation.

Since the proposed method is only functional when sorting populations with two objectives, a generalization of the proposed technique is currently being developed which should improve NSGA-II's sorting method for populations with any number of objectives.

References

[1] H. A. Abbass, S. Baker, A. Bender and R. Sarker, “Identifying the fleet mix in a military setting,” The Second International Intelligent Logistics Systems Conference, 22 - 23 February 2006, Brisbane, Queensland.

[2] S. Baker, D. Morton, R. Rosenthal and L. Williams, “Optimizing military airlift,” Operations Research 50(4):582–602.

[3] S. Wesolkowski and Z. Zhu, “Optimizing the Stochastic Fleet Estimation Model Using Evolutionary Computation,” IEEE World Congress on Computational Intelligence, Hong Kong, China, 2008.

[4] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons: Chichester, UK, 2001.

[5] K. Deb, A. Pratab, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Trans. Evol. Comput., vol. 6, pp. 182-197, Apr. 2002.

[6] Knuth, Donald (1998). “The Art of Computer Programming”. Addison-Wesley: Reading, Massachusetts, p. 158-168.

This page intentionally left blank.

List of symbols/abbreviations/acronyms/initialisms

CFAWC	Canadian Forces Aerospace Warfare Centre
DRDC	Defence Research & Development Canada
NDF	Non-dominated Front
NDS	Non-dominated Set
NSGA-II	Non-dominated Sorting Genetic Algorithm - II
R&D	Research & Development

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

   Canadian Forces Aerospace Warface Centre (CFAWC)
   OR Team
   Centre for Operational Research and Analysis
   Defence R&D Canada
   3701 Carling Avenue
   Ottawa, ON K1A 0K2
   Canada

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

   UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

   Non-dominated Sorting on Two Objectives

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

   Mazurek, M.; Wesolkowski, S.B.

5. DATE OF PUBLICATION (Month and year of publication of document.)

   July 2009

6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)

   16

6b. NO. OF REFS (Total cited in document.)

   6

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

   Technical Note

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

   Canadian Forces Aerospace Warface Centre (CFAWC) OR Team
   Trenton CFB
   P.O. Box 1000 Station Forces
   Astra, ON K0K 3W0

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

   DRDC CORA TN 2009-027

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

   Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

   Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

The fast and elitist non-dominated sorting genetic algorithm (NSGA-II) contains a mechanism to sort individuals in a multi-objective optimization problem into non-dominated fronts, based on their performance in each optimization variable. When dealing with a bi-objective problem it is possible to carry-out the non-dominated sorting more efficiently, using the new sorting method presented in this paper. Results comparing the efficiency of the two sorting mechanisms on sets of data with two objectives are presented and discussed.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Non-dominated Sorting; Multi-objective Optimization; NSGA-II

DEFENCE R&D DÉFENSE

DRDC CORA

www.drdc-rddc.gc.ca