Yingchareonthawornchai, Sorrachai; Roy, Proteek Chandan; Laekhanukit, Bundit; Torng, Eric; Deb, Kalyanmoy

Worst-case conditional hardness and fast algorithms with random inputs for non-dominated sorting

Published in:
GECCO 2020 Companion - Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion

DOI:
10.1145/3377929.3390073

Published: 08/07/2020

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Yingchareonthawornchai, S., Roy, P. C., Laekhanukit, B., Torng, E., & Deb, K. (2020). Worst-case conditional hardness and fast algorithms with random inputs for non-dominated sorting. In *GECCO 2020 Companion - Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion* (pp. 185-186). ACM. https://doi.org/10.1145/3377929.3390073

Worst-case Conditional Hardness and Fast Algorithms with Random Inputs for Non-dominated Sorting *

Sorrachai Yingchareonthawornchai
Aalto University, Finland
sorrachai.yingchareonthwornchai@aalto.fi

Proteek Chandan Roy
Michigan State University, USA
royprote@msu.edu

Bundit Laekhanukit
Shanghai University of Finance and Economics, China
bundit@sufe.edu.cn

Eric Torng
Michigan State University, USA
torng@msu.edu

Kalyanmoy Deb
Michigan State University, USA
kdeb@egr.msu.edu

ABSTRACT

We study the computational complexity of the non-dominated sorting problem (NDS): Given a set P of n points in $\mathbb{R}^m$, for each point $p \in P$, compute ℓ, the length of longest domination chain $p_1 \succ p_2 \succ \cdots \succ p_\ell = p$, where x dominates y (denoted as $x \succ y$) if x is not larger than y in every coordinate. A special case of NDS, which we label as NDS1, is to find all the non-dominated points in P. NDS has emerged as a critical component for multi-objective optimization problems (MOPs). For $m \leq 3$, $\Theta(n \log n)$-time is known. For a fixed small $m > 3$, the best bound is $O(n \log^{m-2} n \log \log n)$. For larger m, the best result is an $O(mn^2)$-time algorithm.

We show that the $O(mn^2)$ running time is nearly optimal by proving an almost matching conditional lower bound: for any $\epsilon > 0$, and $\omega(\log n) \leq m \leq (\log n)^{O(1)}$, there is no $O(mn^{2-\epsilon})$-time algorithm for NDS or NDS1 unless a popular conjecture in fine-grained complexity theory is false. To complete our results, we present an algorithm for NDS with an expected running time $O(mn + n^2/m + n \log^2 n)$ on uniform random inputs.

CCS CONCEPTS

• **Theory of computation → Algorithm design techniques**;

ACM Reference Format:
Sorrachai Yingchareonthawornchai, Proteek Chandan Roy, Bundit Laekhanukit, Eric Torng, and Kalyanmoy Deb. 2020. Worst-case Conditional Hardness and Fast Algorithms with Random Inputs for Non-dominated Sorting . In *Genetic and Evolutionary Computation Conference Companion (GECCO '20 Companion), July 8–12, 2020,Cancun, Mexico.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3377929.3390073

*Partially supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 759557, Science and Technology Innovation 2030 -"New Generation of Artificial Intelligence" Major Project No.(2018AAA0100903), NSFC grant 61932002, Program for Innovative Research Team of Shanghai University of Finance and Economics (IRTSHUFE) and the Fundamental Research Funds for the Central Universities. We also thank Karthik C. S. for insightful discussion.

GECCO '20 Companion, July 8–12,2020, Cancun, Mexico

ACM ISBN 978-1-4503-7127-8/20/07.
https://doi.org/10.1145/3377929.3390073

1 INTRODUCTION AND RELATED WORK

We study the computational complexity of the non-dominated sorting problem (NDS). Let P be a set of n points in $\mathbb{R}^m$. We say that a point p *dominates*[1] another point q, denoted by $p \succ q$, if $p_i \leq q_i$ for positive $i \leq m$, i.e., p is no larger than q in every coordinate and $p \neq q$. Note that it is possible that neither p nor q dominates the other point. A point p is *non-dominated* w.r.t. P if p is not dominated by any other points in P. Given a set of P points, the non-dominated sorting problem asks to compute the *rank function* $R : P \to \mathbb{N}$ defined as follows: $R(p) = 1$ if p is a non-dominated point, and $R(p) = 1 + \max\{R(q) : q \succ p\}$, otherwise. A widely used variant of NDS, which we label NDS1, is to find all $p \in P$ such that $R(p) = 1$.

Non-dominated sorting has emerged as a critical component for multi-objective optimization problems (MOPs). In contrast to single objective optimization where we try to find the best possible solution, the desired result of an MOP is typically a set of Pareto-optimal solutions that reflect the trade-offs among different objectives. An NDS algorithm is a computational bottleneck for multi- and many-objective evolutionary algorithms (MOEAs). NDS1 has broader applications beyond MOEAs, notably including skyline query in database [6].

The non-dominated sorting problem is completely solved when $m = 2$ or 3 with a worst-case time complexity of $\Theta(n \log n)$ [7, 9]. For a fixed $m > 2$, $O(n \log^{m-1} n)$-time algorithms are known using divide-and-conquer (D&C), often referred to as Jensen's sort [5], and the best known running time for this approach is $O(n \log^{m-2} n \log \log n)$-time algorithm [1]. For general m, the first $O(mn^2)$-time algorithm is due to Deb *et al.* [2]. Since then there have been several algorithms achieving the same worst-case bounds, but focusing on practical running time [3, 4, 8, 10, 11, 13]. Until now, the $O(mn^2)$-time bound has stood for almost two decades. Further note that NDS1 does not have a known faster solution.

1.1 Our Results

We show that the running time $O(mn^2)$ is essentially optimal assuming the *Hitting Set Conjecture* (HSC) is true. The Hitting Set Problem (HS) is defined as follows: Given two families of sets A and B containing n sets each over the universe $\{1, \ldots, m\}$ where

[1] we use notion of $p_i \leq q_i$ to be consistent with the MOEA community where, in the context of minimization, the point p is better when every coordinate is smaller than q.

$m = \omega(\log n)$, decide if there exists a set $a \in A$ that intersects (hits) every set $b \in B$ in at least one element. If we use binary vectors to represent sets, the HS problem is equivalently defined as follows: Given two sets A and B of vectors over $\{0,1\}^m$ where $m = \omega(\log n)$, decide if there exists a vector $a \in A$ such that for all $b \in B$, $\sum_{i=1}^{m} a_i \cdot b_i > 0$.

Hitting Set Conjecture (HSC). For any constant $\epsilon > 0$, and m where $\omega(\log n) \leq m < (\log n)^{O(1)}$, there is no $O(mn^{2-\epsilon})$-time algorithm for the Hitting Set Problem.

To date, several hypotheses have been used to prove conditional lower bounds of "easy" problems, e.g., the *All-Pairs Shortest Paths* (APSP) conjecture, the *Orthogonal Vectors* (OV) conjecture, and the *Hitting Set* (HS) conjecture; please see [12] and references therein. We add to this growing body of fine-grained complexity results by using the HSC to prove our main result.

Theorem 1.1. *For any constant $\epsilon > 0$, and m where $\omega(\log n) < m < (\log n)^{O(1)}$, there is no $O(mn^{2-\epsilon})$-time algorithm for NDS or NDS1 unless the HSC is false.*

While our result, assuming the HSC is true, rules out any chance of significantly improving the $O(mn^2)$ worst case bound, there are ways to get better results. We list two here. First, there may exist an algorithm with worst case running time $O\left(mn^2/\log^{O(1)} n\right)$. Second, algorithms may perform better on restricted input instances. We now give our second result which exploits the second option.

Theorem 1.2. *Under the uniform random input assumption, there is an algorithm that takes as input a set of points P in $\mathbb{R}^m$, and outputs the rank function (as defined in the NDS problem) in expected time $O(n^2/m + mn + n\log^2 n)$.*

Proof sketch. The algorithm is a variant of the Best Order Sort (BOS) algorithm from [10]. The key intuition is that for random inputs, the expected running time of BOS improves by roughly an m^2 factor for two reasons. First, most points are mutually non-dominated, and verifying this takes only $O(1)$ (instead of m) expected time. Second, all points are ranked after exploring $O(n/m)$ (instead of n) expected points. □

2 CONDITIONAL HARDNESS OF NON-DOMINATED SORTING

To prove Theorem 1.1, we introduce the following problem.
Bichromatic Binary Non-Dominating Problem (BBND). Given two sets of points A and B where (i) $|A| = |B| = n$ and (ii) each point is a vector over $\{0,1\}^m$ where $m = \omega(\log n)$, decide if there exists a point $a \in A$ that is not dominated (i.e., there exists a positive $i \leq m$ such that $a_i < b_i$) by any point $b \in B$.

Lemma 2.1 (Reduction from HS to BBND). *If BBND can be solved in time $T(m,n)$, then HS can be solved in time $O(T(m,n) + mn)$.*

Proof. We define a reduction $R(A,B) = (A',B)$ from HS to BBND using the binary vector formulation of the HS problem as follows. Set $A' := \{a' : a \in A\}$ where $a'_i = 1$ if $a_i = 0$, and $a'_i = 0$ if $a_i = 1$ for $i \leq m$. Clearly R runs in time $O(mn)$.

We now prove the completeness of R. Assume that (A,B) is a yes-instance to HS. That is, there exists $a \in A$ such that for all $b \in B$, $\sum_{i=1}^{m} a_i \cdot b_i > 0$. This means there is a positive $i \leq m$ such that $a_i = b_i = 1$. By construction (since we flip bitwise from a to a'), for the same i, we have $a'_i = 0$ and $b_i = 1$, so $a'_i < b_i$. This holds for every $b \in B$. Therefore, this particular vector $a' \in A'$ is not dominated by any $b \in B$, so (A',B) is a yes-instance for BBND

We now prove the soundness of R. Assume that (A',B) is a yes-instance to BBND. So, there is a point $a' \in A'$ that is not dominated by any point $b \in B$. This means for each $b \in B$, there is a positive $i \leq m$ such that $a'_i < b_i$. Since a' and b are binary vectors, this implies $a'_i = 0$ and $b_i = 1$. By construction, we have $a_i = 1$ and $b_i = 1$ for the same i. This holds for every $b \in B$. Hence, this vector a has the property that $\sum_{i=1}^{m} a_i \cdot b_i > 0$ for all $b \in B$, so (A,B) is a yes-instance for HS. □

Next we solve $BBND$ using an NDS1 algorithm. The following lemma is obvious.

Lemma 2.2. *The input (A,B) for BBND is a yes-instance if and only if there exists $a \in A$ such that $R_{A\cup B}(a) = 1$ in the solution population $A \cup B$ where $R_{A\cup B}(p)$ is the rank of a solution p in the population $A \cup B$.*

We now prove Theorem 1.1.

Proof of Theorem 1.1. By Lemma 2.1, it is enough to solve the BBND problem in time $O(mn^{2-\epsilon})$. If there is an $O(mn^{2-\epsilon})$-time algorithm for NDS1, we solve BBND in time $O(mn^{2-\epsilon})$ as follows: Given an instance (A,B), we find all the elements of $A \cup B$ with rank 1 and output yes if there is an $a \in A$ in this set and no otherwise. The correctness follows immediately from Lemma 2.2. □

REFERENCES

[1] Maxim Buzdalov. 2019. Make Evolutionary Multiobjective Algorithms Scale Better with Advanced Data Structures: Van Emde Boas Tree for Non-dominated Sorting. In *EMO (Lecture Notes in Computer Science)*, Vol. 11411. Springer, 66–77.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on* 6, 2 (Apr 2002), 182–197.

[3] Hongbing Fang, Qian Wang, Yi-Cheng Tu, and Mark F. Horstemeyer. 2008. An Efficient Non-dominated Sorting Method for Evolutionary Algorithms. *Evol. Comput.* 16, 3 (Sept. 2008), 355–384.

[4] Patrik Gustavsson and Anna Syberfeldt. 2018. A New Algorithm Using the Non-Dominated Tree to Improve Non-Dominated Sorting. *Evolutionary Computation* 26, 1 (2018).

[5] M.T. Jensen. 2003. Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms. *Evolutionary Computation, IEEE Transactions on* 7, 5 (Oct 2003), 503–515.

[6] Christos Kalyvas and Theodoros Tzouramanis. 2017. A Survey of Skyline Query Processing. *CoRR* abs/1704.01788 (2017).

[7] Sanjiv Kapoor and Prakash V. Ramanan. 1989. Lower Bounds for Maximal and Convex Layers Problems. *Algorithmica* 4, 4 (1989), 447–459.

[8] Kent McClymont and Ed Keedwell. 2012. Deductive Sort and Climbing Sort: New Methods for Non-dominated Sorting. *Evol. Comput.* 20, 1 (March 2012), 1–26.

[9] Yakov Nekrich. 2011. A Fast Algorithm for Three-Dimensional Layers of Maxima Problem. In *WADS (Lecture Notes in Computer Science)*, Vol. 6844. Springer, 607–618.

[10] Proteek Chandan Roy, Md. Monirul Islam, and Kalyanmoy Deb. 2016. Best Order Sort: A New Algorithm to Non-dominated Sorting for Evolutionary Multiobjective Optimization. In *GECCO (Companion)*. ACM, 1113–1120.

[11] Handing Wang and Xin Yao. 2014. Corner Sort for Pareto-Based Many-Objective Optimization. *Cybernetics, IEEE Transactions on* 44, 1 (Jan 2014), 92–102.

[12] Virginia Vassilevska Williams. 2015. Hardness of Easy Problems: Basing Hardness on Popular Conjectures such as the Strong Exponential Time Hypothesis (Invited Talk). In *IPEC (LIPIcs)*, Vol. 43. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 17–29.

[13] X. Zhang, Y. Tian, R. Cheng, and Y. Jin. 2018. A Decision Variable Clustering-Based Evolutionary Algorithm for Large-Scale Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* 22, 1 (Feb 2018), 97–112.