Special Issue in Honor of the 60th Birthday of Professor Imre J. Rudas

Foreword

This special issue of *Acta Polytechnica Hungarica* is dedicated to Professor Imre J. Rudas on the occasion of his 60th birthday and in recognition of his outstanding contributions to computational intelligence, robotics with special emphasis on robot control, soft computing, computer-aided process planning, fuzzy control and fuzzy sets. The issue can be considered as a complement to the edited volume *J. Fodor and J. Kacprzyk (Eds.), Aspects of Soft Computing, Intelligent Robotics and Control (Studies in Computational Intelligence, vol. 241) (Springer-Verlag, Berlin, Heidelberg, 2009).* The manuscripts were submitted in response to direct invitations from the Editor-in-Chief, containing contributions from some of Imre's past collaborators. The Editor-in-Chief is grateful to the authors for their excellent work.

It is impossible to summarize in a short foreword the accomplishments of the honoree. At the age of 60 he is an author of books, book chapters, journal papers, conference proceedings papers, all together over 460 scientific publications in the mentioned fields.

Professor Imre J. Rudas graduated from Bánki Donát Polytechnic, Budapest in 1971, received the Master Degree in Mathematics from the Eötvös Loránd University, Budapest, the Ph.D. in Robotics from the Hungarian Academy of Sciences in 1987, while the Doctor of Science degree from the Hungarian Academy of Sciences in 2004. He received his first Doctor Honoris Causa degree from the Technical University of Košice, Slovakia and the second one from "Politehnica" University of Timisoara, Romania.

He served as the Rector of Budapest Tech from August 1, 2003 for a period of four years, and was reelected for five years in 2007. He is active as a full university professor and Head of Department of Intelligent Engineering Systems. During his leadership period Budapest Tech has succeeded in fulfilling the requirements for university status, and ready to work as Óbuda University from January 1, 2010.

Imre is a Fellow of IEEE, Senior Administrative Committee member of IEEE Industrial Electronics Society, member of Board of Governors of IEEE SMC Society and Chair of IEEE Hungary Section. He is the Vice-President of IFSA (International Fuzzy System Association), he was the President of Hungarian

Fuzzy Association for ten years, Steering Committee Member of the Hungarian Robotics Association and the John von Neumann Computer Society.

He serves as an associate editor of diverse scientific journals, member of various national and international scientific committees. He is the founder of the IEEE International Conference Series on Intelligent Engineering Systems (INES) and IEEE International Conference on Computational Cybernetics (ICCC), and some regional symposia. He has served as General Chairman and Program Chairman of numerous scientific international conferences.

His present areas of research activity are Computational Cybernetics, Robotics with special emphasis on Robot Control, Soft Computing, Computer-aided Process Planning, Fuzzy Control and Fuzzy Sets. He has published books, more than 400 papers in books, various scientific journals and international conference proceedings.

Imre J. Rudas' personal qualities of commitment, integrity, leadership and initiate leave lasting impression on his colleagues, students and friends. This issue, which is an expression of their appreciation and friendship, is the 21st one from 2004 may be a surprise and bring a lot of success and satisfaction to him, who launched the life of the journal five years ago.

We wish Imre the best of health, energy and happiness in the years ahead, and we count on his activities and contributions to scientific projects for many years to come.

János Fodor

Editor-in-Chief

Robustness versus Performance in Sorting and Tournament Algorithms

Wilfried Elmenreich, Tobias Ibounig, István Fehérvári

Mobile Systems/Lakeside Labs
University of Klagenfurt, Austria
wilfried.elmenreich@uni-klu.ac.at
ibounig@lakeside-labs.com
istvan.fehervari@uni-klu.ac.at

Abstract: In this paper we analyze the robustness of sorting and tournament algorithms against faulty comparisons. Sorting algorithms are differently affected by faulty comparisons depending on how comparison errors can affect the overall result. In general, there exists a tradeoff between the number of comparisons and the accuracy of the result, but some algorithms like Merge Sort are Pareto-dominant over others. For applications, where the accuracy of the top rankings is of higher importance than the lower rankings, tournament algorithms such as the Swiss System are an option. Additionally, we propose a new tournament algorithm named Iterated Knockout Systems which is less exact but more efficient than the Swiss Systems.

Keywords: sorting algorithms, robustness, tournaments, iterated knockout system

1 Introduction

Sorting is a fundamental and often applied algorithm in computer science. There has been put much attention on the efficiency of a sorting algorithm in terms of number of comparisons or number of element switches. In some applications, like sports tournaments or comparative evolutionary algorithms, the comparison function is a complex function that involves either a match between two players or a simulation of two teams trying to achieve a given goal [1]. In such cases, especially when the opponents are of similar strenght, the outcome of a comparison can become indeterministic (e.g., due to a sports team winning over a stronger team by being lucky).

Our motivation is thus to research sorting algorithms for these potentially failing comparisons. We expect to have faulty comparisons due to random fluctuations in the evaluation function that compares two elements. With respect to randomly occurring errors, we do not assume a hard limit on the number of occurring faulty

comparisons. Instead we are looking for a method that is *robust* against such problems, that is we allow a deviation from the perfect result, but the result should be gracefully degrading based on the number of faults.

The related work on this topic goes back to the 1960s (Section 2) and shows that the problem has been identified in different fields such as mathematics, computer science, and organizers of social studies or (chess) tournaments.

In our work we examine the robustness of typical sorting and tournament algorithms with respect to faulty comparisons. A key hypothesis was that a very efficient (i.e., low complexity order) sorting algorithm might be more susceptible to errors from imprecise comparisons than the more inefficient sorting algorithms which might implement a lot of implicitly redundant comparisons. While our results from an expreimental validation of several standard sorting algorithms in general support this hypothesis, there are still some intrinsic factors in the way of sorting that make an algorithm more or less robust to these faults. We show that therea is a tradeoff between accuracy and number of comparisons and place the results for Bubble Sort, Selection Sort, Heap Sort, Quick Sort, Merge Sort and Insertion Sort on a two-dimensional map of both criteria. As shown in Secion 6, Merge Sort provides a good accuracy for a reasonable comparion overhead. Additionally, we have examined tournament systems such as Round Robin, Swiss System and propose an Iterated Knockout System (IKOS). These algorithms are especially of interest for sorting tasks where the accuracy of the topmost places is the most important while errors in the lower ranks do not play a role. For this case, IKOS has shown the highest efficiency.

The insights gained from this work (Section 7) may be a helping guideline for selecting a sorting algorithm under noisy conditions. In particular they are useful for implementing a fair but time-efficient tournament that determines the best teams. Another application can be in evolutionary algorithms with comparative fitness functions, as for example in [1]. Since the fitness comparison often requires a time-consuming simulation, cutting down on the number of comparisons (i.e., simulation runs) while keeping the accuracy for the upper part of the population is an important issue.

2 Related Work

There exists a vast amount of literature on sorting algorithms [2, 3]. In the following we review work where the problem on robustness and fault tolerance is particularly treated.

Binary search with faulty information was formulated as game theoretic problem by Rényi [4] (a player must guess an object based on yes/no answers from another player that sometimes may answer incorrectly) and by Ulam [5] (a very similar

game where a player must guess a number in a given range). A more comprehensive overview on the historical development of research on this topic can be found in [6].

Approximate voting is also supported by several algorithms supporting anytime behavior, i.e., a process generations intermediate results which increase in their accuracy over time. An example for such an algorithm is Comb Sort [7], which iteratively "combs" the elements similar to a bubble-sort approach. However, the comb sort provides adjustment for specifically focussing on a specific part (e.g. the first few positions) of the list to be sorted.

Ravikumar, Ganesan, and Lakshmanan discuss the problem of finding the largest element of a set using imperfect comparisons [8]. The approach is extended in [9] to an algorithm for sorting elements with a comparison function that may sometimes fail, i.e., yielding the incorrect results. The algorithm has a worst-case complexity of $\Omega(n\,log(n) + e\,n)$, where e is an upper bound for the total number of errors. Based on these results, Long [10] presents an algorithm for searching and sorting with a faulty comparison oracle. Given that the assumption on e does hold, these algorithms provide a perfect ranking. However, there is no assessment on the sorting quality if this assumption is invalidated. Thus, these algorithms are fault-tolerant, but not necessarily robust.

Bagchi presents a similar approach in [11]. His fault-tolerant algorithm is basically a binary insertion sort modified to cope with errors also with a worst-case complexity of $\Omega(n\,log(n) + e\,n)$.

Ajtai et al. [12] assume a different model for imprecise comparisons, where the outcome of a comparison is considered unpredictable if the elements differ by less than a given threshold δ. They present an algorithm that provides a correct sorting of all elements which differ at least by δ.

Giesen et al. [13] present a worst-case bound for the necessary comparisons of any approximate sorting algorithm (however without cosidering faulty comparisons) that ranks n items within an expected Spearman's Footrule distance.

3 Why Robustness

In contrast to the well-established and well-defined field of fault tolarence [14, 15], the notion of robustness differs by the field of research [16]: *"A biologist will understand robustness in terms like adaptation, stability, diversity, survivability, and perturbations. A control theorist will express robustness in terms of uncertainty of mathematical models. A software developer might focus on a programs ability to deal with unusual usage or users input."*

Robustness differs from fault tolerance in the way that robustness is not implemented against a rigid fault hypothesis [17].

Instead, robustness (against a particular property, such as noisy sensor data, etc.) points out that the system is capable of maintaining its function (at least in a degraded, but acceptable way) despite various unexpected perturbations.

When applying the concept of robustness for sorting, we are looking for algorithms that might degrade in its result, i.e., the sorting order, but provide an approximate result which is acceptable. Although we can use standard measures to define if a result is more or less deviating from the correct sorting, the level of acceptability heavily depends on the application. The application we had in mind was performing a sorting of candidates in a genetic algorithm where the fitness function is inaccurate [1]. In genetic algorithms, an inaccurate sorting is likely to have only a degrading effect on the runtime of the algorithm, i.e., slowing down the convergence of the gene pool towards a solution with high fitness. Hence, there exist mutual tradeoffs between sorting speed/sorting accuracy and sorting accuracy/speed of genetic algorithms.

For the sake of generality, we will analyze several sorting approaches yielding different combinations of sorting performance and accuracy.

4 Algorithms under Consideration

4.1 Sorting Algorithms

As a first step we will evaluate a number of standard sorting algorithms. We have selected Bubble Sort[1], Selection Sort[2], Insertion Sort[3], Heap Sort[4], Quick Sort (using a simple randomized function to define the pivot) [5], and Merge Sort[6] for our test. The first three sorting algorithms are in the complexity order of $O(n^2)$, i.e., they are typically very inefficient for a high number of elements. The other three algorithms are in the complexity order $O(n \log n)$, thus more efficient.

The majority of the sorting algorithms considered in this paper are symmetric towards sorting the whole set in similar quality. An exception is the Heap Sort

1 http://en.wikipedia.org/wiki/Quicksort
2 http://en.wikipedia.org/wiki/Selection_sort
3 http://de.wikipedia.org/wiki/Insertionsort
4 http://en.wikipedia.org/wiki/Heapsort
5 http://en.wikipedia.org/wiki/Quicksort
6 http://en.wikipedia.org/wiki/Merge_sort

algorithm and the Swiss system. Under the presence of faulty comparions, Heap Sort turned out to make more sorting errors in the top ranks rather than in the last ranks. Therefore, we inverted the Heap Sort algorithm in order to have the better sorting in the top ranks.

4.2 Tournament Algorithms

The sorting problem is very similar to the task of organizing a tournament among a number of participants. Participants are paired into matches deciding which participant is stronger and should therefore be sorted "above" the other one.

In contrast to sorting, tournament organizers usually consider that comparisons are neither deterministic nor consistently yielding the stronger participant. In the simplest form, the round robin tournament, every participant is paired against every other participant. The results of each match give points to the participants, which are sorted according to their points in the end. Note that for example, a participant could win a tournament even though he or she lost to the second ranked participant.

A round-robin approach is usually very robust against random influences on the comparison function, since the pairing does not depend on the outcome of previous comparisons. However, this approach requires $\frac{n(n-1)}{2}$ comparisons and is thus as inefficient as the sorting algorithms in the complexity order of $O(n^2)$.

The Swiss Systems style tournament [18] is more efficient than the round robin tournament. The Swiss System is extensively used in chess tournaments. When there is no *a priori* knowledge of the participants' strenght, the first round of s Swiss System tournament contains random pairings. In each game the winner gets two points, loser gets zero, in case of a draw both get one point. After this round players are placed in groups according to their score (winners in the group "2", those who drew go in the group "1" and losers go into the group "0"). The aim is to ensure that players with the same score are paired against each other. Since the number of perfect scores is cut in half each round it does not take long until there is only one player remaining with a perfect score. In chess tournaments there are usually many draws, so more players can be handled (a 5 round event can usually determine a clear winner for a section of at least 40 players, possible more).

The drawback of the Swiss system is that it is only designed to determine a clear winner in just a few rounds. Likewise, the worst performing participant is also determined. The more a position differs from the first or last position, the less likely this position is correctly ranked. In other words, the Swiss system has an increasing exactness towards the first few and last few ranks.

4.3 Iterated Knockout System

Some of the applications we had in mind only need a sorting of top half of the elements (e.g., a genetic algorithm that drops all candidates below a threshold). Therefore, we developed a specific algorithm to sort a given number of ranks starting from the "first place":

1 start with an empty ranking list;

2 start a single-elimination tournament: each candidate takes place in exactly one pairing per round. The winners of each pairing promote to the next round. If the number of candidates is uneven, one candidate not being paired passes on to the next round.

3 iterate 2 until there is only one candidate (the winner of this tournament) left;

4 append the winner to the overall ranking list;

5 build the list of candidates (except the ones already ranked) that have not lost to anyone except for the already ranked candidates;

6 go to step 2. Results from already played pairings are kept.

Thus, we subsequently pick players from the list until the ranking list contains all the ranks of interest.

5 Evaluation Method

For an evaluation, we test these algorithms on a set with randomly generated numbers in a range between 0 and 100. Array sizes have been varied between 10 and 200 according to typical target applications. For each comparison operation, a random factor (the "noise") is applied to both values before the comparison operation is performed. A 5% noise means for example that the value used for comparison may vary up to ±5% of the value range (100). Thus, the probability that a comparison may yield an incorrect result is the higher the closer the two values are. The average result of a sorting algorithm under test is compared to the correct sorting, that is without applying the random fluctuation before comparison. For the comparison we apply two metrics: The first one is based on Spearman's footrule as a measure of disarray [19], which is calculated as the sum of absolute differences between the resulting and correct ranks. The results are normalized by the number of elements, thus, the deviation in our results always gives the average distance in ranks of an element to its correct position. In order to account for applications where the correct ranking of the lower ranks is not important, we apply also a different metric that apply a weight of 2 for deviations in the top half, while ranking deviations in the lower half have a weight of zero, thus do not contribute to the metric.

6 Results

Figure 1 visualizes the complexity of the different algorithms with respect to the number of comparions. As expected, we can observe the inefficient ($O(n^2)$) sorting algorithms like Bubble Sort, Selection Sort, and Insertion Sort to require by far the most comparisons to create a ranking. The sorting algorithms Heap Sort, Merge Sort, and Quick Sort are more efficient. The Swiss System needs even less number of comparisons, but the Swiss System is no sorting algorithm since it does not yield a perfect sorting of the result even with perfect comparisons.

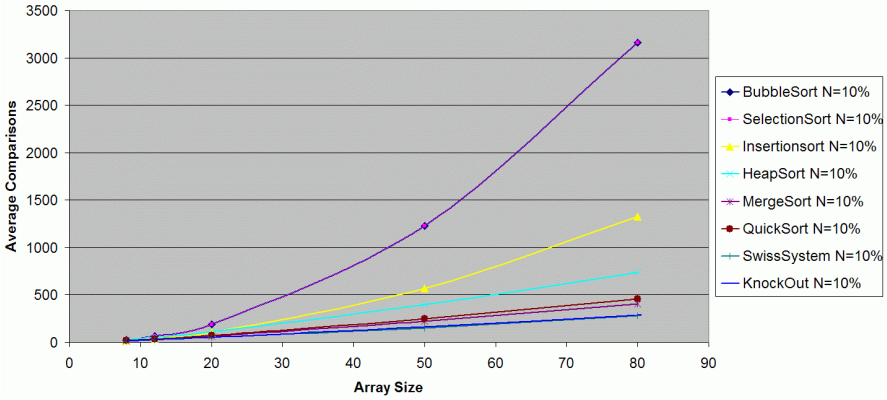


Figure 1
Number of comparisons vs. array size for different algorithms

When considering random fluctuations before comparison, the algorithms show different performances as depicted in Figure 2. The Swiss System, which was the most efficient one in the previous analysis, comes with the cost of hight deviation (disarray according to Spearman's footrule). In this graph, also the performance of a full Round Robin tournament is depicted. In the round robin tournament there are $\frac{n(n-1)}{2}$ comparisons, where each comparison gives a point to the winner. Afterwards, the ranking is established by the number of points. Although not being very efficient, faulty comparisons in the Round Robin tournament approach are likely to cancel out to have their effect limited. Therefore, a Round Robin turns out to be the most robust (but painfully slow) approach. Interestingly, Insertion Sort is both, slow and inaccurate. This is due to the fact that one faulty comparison can affect the ranking of all other elementes and thus leads to subsequent errors.

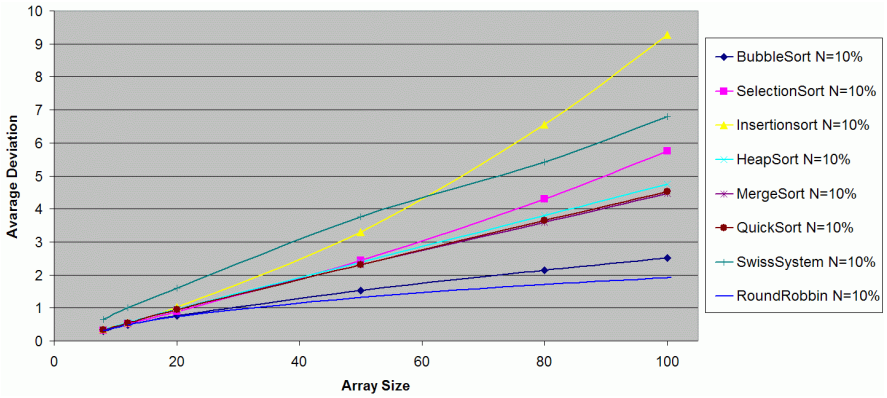


Figure 2
Deviation from the perfect result

Figure 3 examines the robustness of the sorting and tournament approaches for different levels of noise. The array size was chosen to be constant 50. The most robust methods are (sorted according to their robustness): Round Robin, Tournament, Bubble Sort, Merge Sort, Quick Sort, Heap Sort, and Selection Sort. The Swiss System is an interesting case, for low noise levels, it is among the worst methods, however, for noise of 30% and more, the Swiss System is the third best one, since its results degrade slower than the other algorithms.

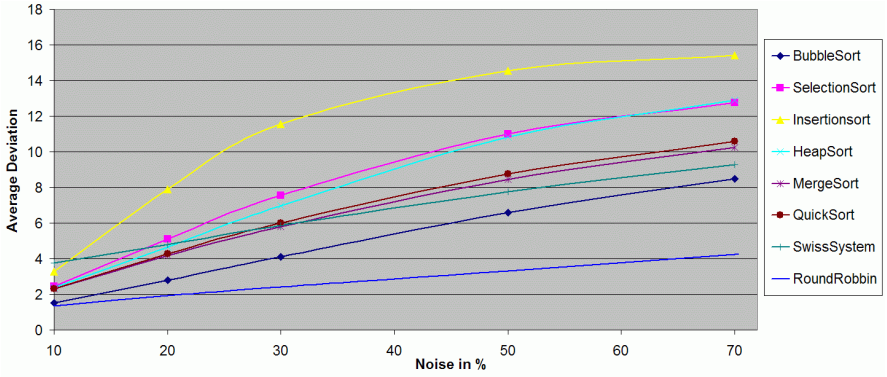


Figure 3
Vulnerability to noise in comparison function

Figure 4 maps the different algorithms according to their average number of comparisons and the average resulting deviation. The noise parameter had been chosen to be 10% and the array size was 50 for that comparison. We observe that Merge Sort dominates Quick Sort, Heap Sort, Selection Sort and Insertion Sort. In other words, Merge Sort is Pareto-optimal among this set. The Round Robin tournament dominates Bubble Sort and Selection Sort. Finally the Swiss system

dominates the IKOS approach (which was set to sort only the upper half). Thus, the algorithms of choice are Round Robin if accuracy is of most importance, Swiss Sort if efficiency is of most importance, and Merge Sort for a combination of both. Quick Sort has only slightly worse results than Merge Sort.

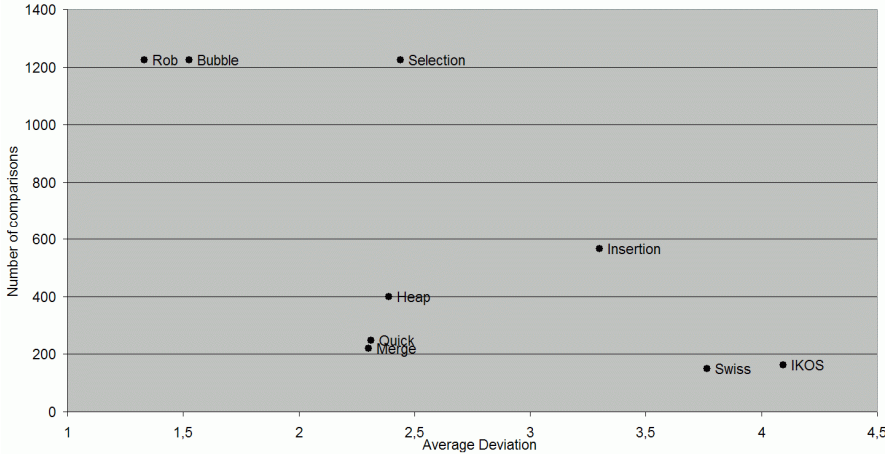


Figure 4

Mapping of different algorithms according to comparison effort and resulting deviation

Figure 5 analyzes the robustness to noise in the comparison function with respect to the top half rank results. Thus, ranking errors in the lower half do not influence the result. Here, our proposed IKOS algorithm shows a better efficiency, since it was designed for this case.

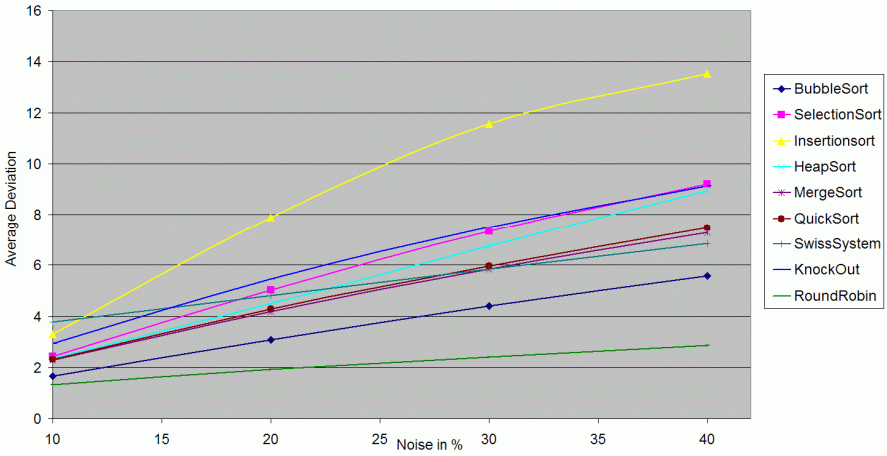


Figure 5

Robustness to noise in top half rank results

Figure 6 depicts the mapping of different algorithms according to comparison effort and resulting deviation in top half rank results. Likewise in the analysis before, the noise parameter had been chosen to be 10% and the array size was 50 for this evaluation.

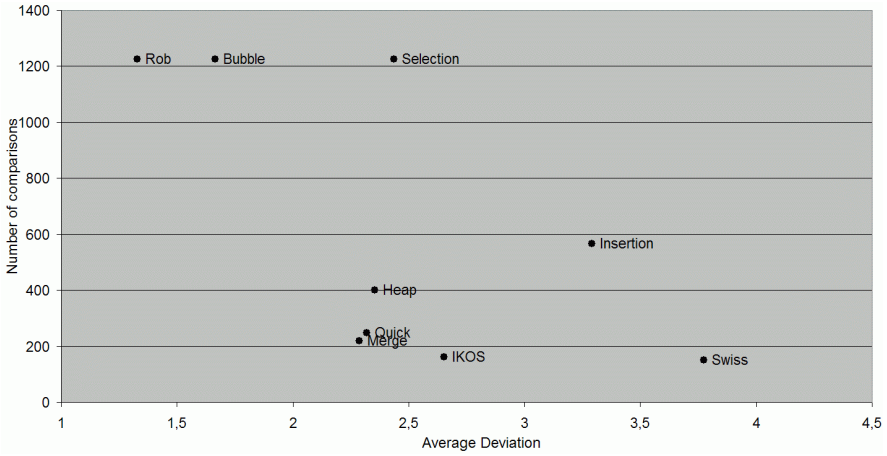


Figure 6

Mapping of different algorithms according to comparison effort and resulting deviation in top half rank results

Again, Round Robin and Merge Sort are Pareto-optimal as before. IKOS and Swiss System, however, switch places. With only slightly more comparisons than Swiss System, IKOS is able to provide a result which less prone to noisy comparions.

Conclusion

This paper contributes in two ways to the state of the art. The first contribution is the analysis of existing sorting algorithms according to their robustness against imprecise or noisy comparisons. In contrast to related work which introduces new algorithms that overcome a defined number of faulty comparisons, our approach did not aim at a fault-free sorting but rather at an approximate sorting with a minimum overhead. This is especially of interest for applications where an expensive and noisy comparison function is used to establish a ranking. If only the ranking of the first few elements is of interest, algorithms designed for (sports) tournaments are an intresting option. Apart from tournaments such a ranking function is of interest for the evaluation phase in genetic algorithms when evolving a comparative fitness function. Therefore, we have also presented a new tournament algorithm that provides an ordering incrementally starting from the top ranks.

The sorting and tournament algorithms under consideration have been evaluated according to their sorting complexity and result accuracy. The results show that

round robin tournament, merge sort, and the Swiss system are Pareto-optimal according to the overall ordering and that round robin tournament, merge sort, and the presented IKOS algorithm are Pareto-optimal according to a sorting of the top half of elements.

The presented algorithms have been selected for their best robustness, but, except for IKOS, have not been especially designed for this case. We think that there is potential for further improving the sorting algorithms by adding mechanisms dedicated to implement robustness.

Acknowledgement

This work was supported by the European Regional Development Fund and the Carinthian Economic Promotion Fund (contract KWF 20214-18128-26673) within the Lakeside Labs project DEMESOS.

References

[1] I. Fehérvári, W. Elmenreich: Evolutionary Methods in Self-Organizing System Design. In *Proceedings of the 2009 International Conference on Genetic and Evolutionary Methods*, 2009

[2] D. E. Knuth: *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, 1997

[3] P. Puschner: Real-Time Performance of Sorting Algorithms. *Real-Time Systems*, 16(1):63-79, January 1999

[4] A. Rényi: On a Problem in Information Theory. *Magyar Tudományos Akadémia Matematikai Kutató Intézet Közlemény*, 6:505-516, 1961

[5] S. M. Ulam: *Adventures of a Mathematician*. Scribner, New York, 1976

[6] A. Pelc: Searching Games with Errors – Fifty Years of Coping with Liars. *Theoretical Computer Science*, 270:71-109, 2002

[7] S. Lacy, R. Box: A Fast, Easy Sort. *Byte Magazine*, p. 315 ff., April 1991

[8] B. Ravikumar, K. Ganesan, K. B. Lakshmanan: On Selecting the Largest Element in spite of Erroneous Information. In *Proceedings of the 4th Annual Symposium on Theoretical Aspects of Computer Science (STACS 87)*, Passau, Germany, 1987, pp. 88-99

[9] K. B. Lakshmanan, B. Ravikumar, K. Ganesan: Coping with Erroneous Information while Sorting. *IEEE Transactions on Computers*, 40(9):1081-1991, September 1991

[10] P. M. Long: Sorting and Searching with a Faulty Comparison Oracle. Technical Report UCSC-CRL-92-15, University of California at Santa Cruz, 1992

[11] A. Bagchi: On Sorting in the Presence of Erroneous Information. *Information Processing Letters*, 43(4):213-215, 1992

[12] M. Ajtai, V. Feldman, A. Hassidim, J. Nelson: Sorting and Selection with Imprecise Comparisons. In *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, 2009, Vol. Part I, pp. 37-48

[13] J. Giesen, E. Schuberth, M. Stojaković: Approximate Sorting. *Fundamenta Informaticae*, XXI:1001-1006, 1977

[14] A. Avizienis: Fault Tolerance, the Survival Attribute of Digital Systems, *Proceedings of the IEEE*, 66(10):1109-1125, October 1978

[15] J.-C. Laprie, J. Arlat, C. Béounes, K. Kanoun: Definition and Analysis of Hardware- and Software-Fault-Tolerant Architectures. *Computer*, 23(7):39-51, July 1990

[16] V. Mikolasek: Dependability and Robustness: State of the Art and Challenges. In *Workshop on Software Technologies for Future Dependable Distributed Systems*, Tokyo, March 2009

[17] H. Kopetz: On the Fault Hypothesis for a Safety-Critical Real-Time System. In *Keynote Speech at the Automotive Software Workshop San Diego (ASWSD 2004)*, San Diego, CA, USA, January 10-12, 2004

[18] FIDE Swiss Rules. Approved by the General Assembly of 1987. Amended by the 1988 and 1989 General Assemblies.

[19] P. Diaconis, R. L. Graham: Spearman Footrule as a Measure of Disarray. *Journal of the Royal Statistical Society*, Series B 39:262268, 1977

Pliant Arithmetics and Pliant Arithmetic Operations

József Dombi

University of Szeged
Árpád tér 2, H-6720 Szeged, Hungary
dombi@inf.u-szeged.hu

Abstract: Fuzzy arithmetic based α -cuts, where the result of the α -cuts represent an interval. The arithmetic can be understand as an interval arithmetic of the α -cuts. Instead of dealing with intervals we are dealing with left and right hand sided soft inequalities which define the interval. We offer a new calculation procedure of arithmetics, when these soft inequalities meet certain properties (i.e. strict monotonously increasing function represent the inequality). We show that the result of linear combinations of linear is also linear and the linear combination of sigmoid is also sigmoid function (i.e. they are closed under linear combination). We give the result of other operation, too. The soft inequalities define an interval by using proper conjunctive and disjunctive operator. We give such operations, too.

Keywords: fuzzy arithmetic, sigmoid function, triangular membership function, membership function, distending function

1 Introduction

The idea the fuzzy quantities could be arithmetically combined according to the laws of fuzzy set theory is due to Zadeh [14]. Soon after, several researchers worked independently along these lines, such as Jain [5], Mizumoto and Tanaka [9, 10], Nahmias [11], Nguyen [12], Dubois and Prade [1]. It was only further on recognized that the mathematics of fuzzy quantities are an application of possibility theory, an extension of interval analysis as well as of the algebra of many-values quantities (Young [13]).

Fuzzy interval extends and updates the overview of Dubois and Prade [2]. Several theoretical details and applications can be found e.g., in monographs of Kaufmann and Gupta [6, 7], and Mares [8]. In 1987, teher was a special issue of Fuzzy Sets and Systems (Dubois and Prade [3]) devoted to the fuzzy intervals domain, and more recently another one has appeared (Fullér and Mesiar [4]).

In real world applications we often need to deal with imprecise quantities. They can be results of measurements or vague statements, e.g. I have about 40 dollars in my pocket, she is approximately 170 cm tall. In arithmetics we can use $a < x$ and $x < b$ inequalites to characterize such quantities, e.g. if I have about 40 dollars then my money is probably more than 35 dollars and less than 45 dollars.

Fuzzy numbers can also be used to represent imprecise quantities. Pliant numbers are created by *softening* the $a < x$ and $x < b$ inequalities, i.e. replacing the crisp characteristic function with two fuzzy membership functions and applying a fuzzy conjunction operator to combine the two functions. We refer to the softened inequalities as *fuzzy inequalities*.

We call the distending function corresponding to the $x < a$ interval the left side of the fuzzy number and denote it as δ_l. Similarly we refer to the distending function corresponding to the $x < b$ interval as the right side of the fuzzy number and denote it as δ_r.

We will use the following terminology: function representing the soft inequality called distending function. The word pliant means flexible instead of using distending we use soft inequalities and additive pliant is when $f_c(x) + f_d(x) = 1$ (at nilpotent operator case) and multiplicative pliant if $f_c(x) f_d(x) = 1$ (at strict monotone operator case).

Naturally one would like to execute arithmetic operations over fuzzy numbers. Fuzzy arithmetic operations are generally carried out using the α -cut method. In Section 2 we propose a new and efficient method for arithmetic calculations. The next two sections discuss the arithmetic operations and their properties for two classes of fuzzy distending functions. Section 3 investigates additive pliant functions, i.e. distending functions represented as lines. Section 4 presents multiplicative pliant functions, i.e. distending functions based on pliant inequalities. Finally, Section 5 examines which conjunction operators are suitable for constructing fuzzy numbers from additive and multiplicative pliant.

2 Fuzzy Arithmetics

Fuzzy arithmetic operations are based on the extension principle of arithmetics. In arithmetics we can find the result of an arithmetic operation by measuring the distance of the operands from the zero point than applying the operation on these distances. Fig. 1 presents this idea in case of addition.

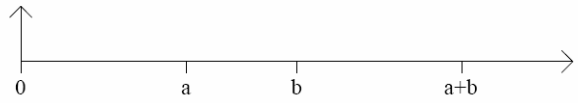


Figure 1
Arithmetic addition

In fuzzy arithmetics we deal with fuzzy numbers. Fuzzy numbers are mappings from real numbers to the $[0,1]$ real interval. Operations are executed by creating an α-cut for all $\alpha \in [0,1]$ and using the arithmetic principle to get the resulting value for each α value. Fig. 2 demonstrates fuzzy addition with fuzzy numbers represented as lines. The dotted triangle number is the sum of the two other triangle numbers.

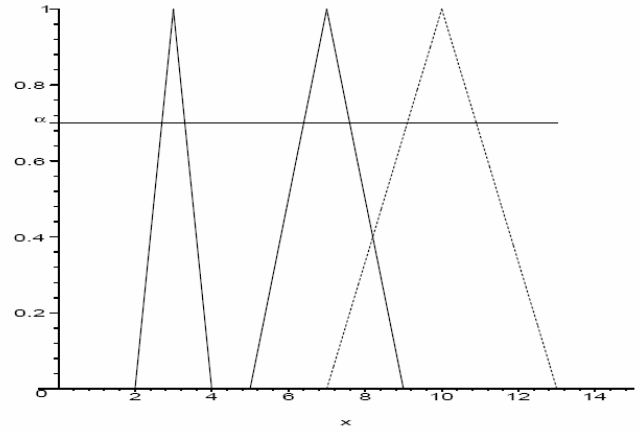


Figure 2
Fuzzy addition with α-cut

This way we can have all the well-known unary ($-x$, x^2) and binary operations ($x+y$, xy, $x \bmod y$) available as fuzzy operations. However the calculation of fuzzy operations with α-cut is tedious and often impractical. In this paper a new efficient method is proposed which is equivalent with the α-cut.

Fuzzy numbers are often composed of two strictly monotone functions, i.e. the left side denoted as δ_l, and the right side denoted as δ_r of the fuzzy number. Fuzzy operations can be carried out by first applying them to the left sides than to the right sides of the operands.

This separation allows us to treat fuzzy numbers as strictly monotone functions when dealing with fuzzy arithmetic operations. In the following we omit the subscript from δ_l and δ_r and simply write δ with the inherent assumption that we shall only do arithmetic operations with functions representing the same side of fuzzy numbers.

Lemma 2.1 *Let* $\delta_1, \delta_2, \ldots, \delta_n$ $(n \geq 1)$ *be strictly monotone functions representing soft inequalities and let* F *be an* n*-ary fuzzy operation over them. If*

$$\delta = F(\delta_1, \delta_2, \ldots, \delta_n),$$

then

$$\delta(z) = \left(F\left(\delta_1^{-1}, \delta_2^{-1}, \ldots, \delta_n^{-1}\right)\right)^{-1}(z) \Leftarrow \tag{1}$$

$$\delta(z) = \sup_{F(x_1, x_2, \ldots, x_n) = z} \min\left\{\delta_1(x_1), \delta_2(x_2), \ldots, \delta(x_n)\right\}.$$

Proof. It can be easily verified that the method is equivalent with the α-cut. □

Fig. 3 visualizes the equivalence for the addition of lines. The left side shows the result of lines added together using α-cut with the result presented as the dotted line. On the right side we have simply added together the inverse functions of the two operands. The result is also presented as a dotted line. It can be seen from the figures that the result of the α-cut is indeed the inverse of the result in the right hand side figure.

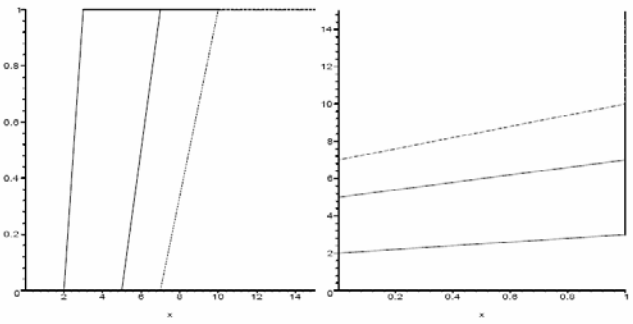


Figure 3
Left: α-cut addition, right: inverse of addition

We can state a theorem regarding the properties of fuzzy operations.

Theorem 2.2 *Let* $\delta_1, \delta_2, \ldots, \delta_n$ $(n \geq 1)$ *be strictly monotone functions representing fuzzy inequalities and let* F *be an* n*-ary fuzzy operation over them. If*

$$F\left(\delta_1^{-1}, \delta_2^{-1}, \ldots, \delta_n^{-1}\right)$$

is strictly monotone then F *has all the properties as its non-fuzzy interpretation.*

Proof. It can be derived from Eq. 1 in Lemma 2.1. □

3 Additive Pliant

Triangle fuzzy numbers are commonly used to represent approximate values. A triangle fuzzy number has one line on each side. We can add triangle fuzzy numbers by first adding their left lines and than adding their right lines together.

Lemma 2.1 let us derive a general formula for adding lines.

Definition 3.1 *We say that a line* $l(x)$ *is given by its mean value if*

$$l(x) = m(x-a) + \frac{1}{2}$$

as shown in Fig. 4.

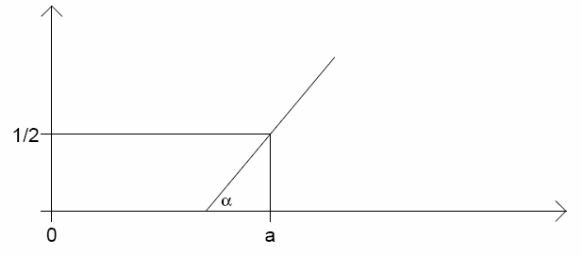


Figure 4
Line given by its mean value a and tangent $m = \tan\alpha$

The inverse of $l(x)$ denoted as $l^{-1}(y)$ can be calculated easily

$$l^{-1}(y) = \frac{y - \frac{1}{2}}{m} + a.$$

3.1 Addition

Theorem 3.2 *Let* $l_i(x) = m_i(x - a_i) + \frac{1}{2}$ *(*$i \in \{1, \ldots, n\}$*) lines given by their mean values. The fuzzy sum of* l_i *lines denoted as* l *is also a line and can be given as*

$$l(x) = l_1(x) \oplus \ldots \oplus l_n(x) = m(x - a) + \frac{1}{2}$$

where

$$\frac{1}{m} = \sum_{i=1}^{n} \frac{1}{m_i} \text{ and } a = \sum_{i=1}^{n} a_i.$$

Proof. Using Lemma 2.1 gives us

$$l^{-1}(y) = \left(l_1^{-1}(y) + \ldots + l_n^{-1}(y)\right)$$

$$= \sum_{i=1}^{n} \left(\frac{y - \frac{1}{2}}{m_i} + a_i \right) = \sum_{i=1}^{n} \left(\frac{y - \frac{1}{2}}{m_i} \right) + \sum_{i=1}^{n} a_i =$$

$$= \left(y - \frac{1}{2} \right) \sum_{i=1}^{n} \frac{1}{m_i} + \sum_{i=1}^{n} a_i$$

From here we have

$$l(x) = \frac{1}{\sum_{i=1}^{n} \frac{1}{m_i}} \left(x - \sum_{i=1}^{n} a_i \right) + \frac{1}{2}.$$

Substituting $\frac{1}{m}$ and a into the equation we get the desired result

$$l(x) = m\left(x - a\right) + \frac{1}{2}.$$

□

3.2 Multiplication by Scalar

Theorem 3.3 *Let*

$$l(x) = m(x-a) + \frac{1}{2}$$

line given by their mean values.

The scalar multiplication of the lines is:

$$c \odot l(x) = m'(x-a') + \frac{1}{2}$$

where

$$a' = ca \qquad m' = \frac{m}{c}$$

Proof. Using Lemma 2.1 gives us

$$l^{-1}(x) = c\left(\frac{y-\frac{1}{2}}{m} + a\right).$$

From here we have

$$l(x) = \frac{m}{c}(x - ca) + \frac{1}{2}.$$

3.3 Subtraction

Calculations for subtraction yields

$$l(x) = l_1 \ominus l_2 = \frac{1}{\frac{1}{m_1} - \frac{1}{m_2}}\left(x - (a_1 - a_2)\right) + \frac{1}{2}.$$

Note: l does not exist when $m_1 = m_2$.

It is an important property that the result of the operation is also a line, i.e. the operation is closed for lines.

3.4 Multiplication and the n^{th} Power

Now let us calculate $\delta = l_1 \otimes l_2$, the product of l_1 and l_2.

$$\delta(y)^{-1} = \left(l_1^{-1} l_2^{-1}\right) = \left(\frac{y-\frac{1}{2}}{m_1} + a_1\right)\left(\frac{y-\frac{1}{2}}{m_2} + a_2\right)$$

$$= \frac{(y-\frac{1}{2})^2}{m_1 m_2} + \left(\frac{a_1}{m_2} + \frac{a_2}{m_1}\right)\left(y-\frac{1}{2}\right) + a_1 a_2$$

$$= \left(\frac{(y-\frac{1}{2})}{\sqrt{m_1 m_2}} + \frac{(\frac{a_1}{m_2} + \frac{a_2}{m_1})\sqrt{m_1 m_2}}{2}\right)^2 - \frac{\left(\frac{a_1}{m_2} + \frac{a_2}{m_1}\right)^2 m_1 m_2}{4} + a_1 a_2$$

$$= \left(\frac{(y-\frac{1}{2})}{\sqrt{m_1 m_2}} + \frac{\frac{1}{2}\left(m_1 a_1 + m_2 a_2\right)}{\sqrt{m_1 m_2}}\right)^2 - \frac{\left(m_1 a_1 + m_2 a_2\right)^2}{4 m_1 m_2} + \frac{4 m_1 a_1 m_2 a_2}{4 m_1 m_2}$$

$$= \frac{\left(y + \frac{1}{2}\left(m_1 a_2 + m_2 a_2\right) - \frac{1}{2}\right)^2}{m_1 m_2} - \frac{\frac{1}{4}\left(m_1 a_1 - m_2 a_2\right)^2}{m_1 m_2}.$$

From here we have

$$\delta(x) = \sqrt{m_1 m_2 x + \frac{1}{4}\left(m_1 a_1 - m_2 a_2\right)^2} - \frac{1}{2}\left(m_1 a_1 + m_2 a_2\right) + \frac{1}{2}.$$

Fig. 5 shows the result of multiplying two lines. The parameters were $m_1 = \frac{1}{5}, a_1 = 4$ and $m_2 = \frac{1}{3}, a_2 = 6$. The multiplication function is shown as the dotted curve.

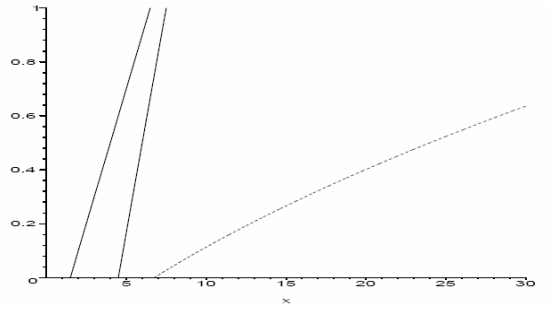


Figure 5
Multiplication of lines

The result is not a line. We need to remain in the world of lines to be able to carry out further arithmetic operations. This can be achieved by approximating the result with a line, e.g. using the least squares method.

Let us calculate the n^{th} power of $l(x) = m(x-a) + \frac{1}{2}$.

$$\delta(y)^{-1} = \left(l^{-1}(y)\right)^n = \left(\frac{y - \frac{1}{2}}{m} + a\right)^n.$$

From here we get

$$\delta(x) = m\left(\sqrt[n]{x} - a\right) + \frac{1}{2}.$$

An approximation method should also be used here to get a line function from the result.

3.5 Properties of Operations

Theorem 3.4 *Addition is commutative and associative over lines.*

Proof. The properties can be easily seen from the construction of $\frac{1}{m}$ and a in Theorem 3.2. □

Theorem 3.5 *Multiplication over lines is commutative, associative and distributive over addition.*

Proof. Theorem 2.2 guarantees that these properties holds. □

4 Multiplicative Pliant

Let us start by introducing a special fuzzy inequality, the *pliant inequality* and examine its most important properties.

4.1 Pliant Inequality Model

Definition 4.1 *A pliant inequality is given as a sigmoid function of*

$$\{a <_\lambda x\} = \frac{1}{1+e^{-\lambda(x-a)}} = \sigma_a^{(\lambda)}(x)$$

where a *is the mean value, i.e.* $\sigma_a^{(\lambda)}(a) = \frac{1}{2}$.

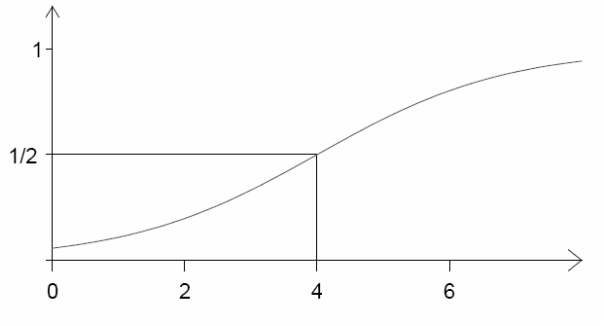


Figure 6
Pliant inequality with $\lambda = 0.7$ and $a = 4$ parameters

The following properties can be seen from the figure

$$a < x \text{ then } \{a <_\lambda x\} > \frac{1}{2},$$

$$a = x \text{ then } \{a <_\lambda a\} = \frac{1}{2},$$

$$a > x \text{ then } \{a <_\lambda x\} < \frac{1}{2}.$$

Definition 4.2 *The inverse function of* $\sigma_a^{(\lambda)}(x)$ *is denoted as* $\left(\sigma_a^{(\lambda)}\right)^{-1}(x)$ *and can be calculated easily. Let*

$$\sigma_a^{(\lambda)}(x) = \frac{1}{1+e^{-\lambda(x-a)}} = \omega,$$

then

$$1 = \omega(1+e^{-\lambda(x-a)}) = \omega + \omega e^{-\lambda(x-a)}$$

$$\frac{1-\omega}{\omega} = e^{-\lambda(x-a)}$$

$$\ln\left(\frac{1-\omega}{\omega}\right) = -\lambda(x-a)$$

$$x = \left(\sigma_a^{(\lambda)}\right)^{-1}(\omega) = \frac{1}{-\lambda}\ln\left(\frac{1-\omega}{\omega}\right) + a.$$

Definition 4.3 *The first derivative of* $\sigma_a^{(\lambda)}(x)$ *is denoted as* $\left(\sigma_a^{(\lambda)}(x)\right)$. *The following properties hold*

$$\left(\sigma_a^{(\lambda)}(a)\right) = \frac{d\sigma_a^{(\lambda)}(x)}{dx}|_{x=a} = 4\lambda;$$

depending on λ, *if*

$\lambda > 0$ *then* $\left(\sigma_a^{(\lambda)}(x)\right)'$ *is strictly monotone increasing,*

$\lambda = 0$ *then* $\left(\sigma_a^{(\lambda)}(x)\right)' \equiv 0,$

$\lambda < 0$ *then* $\left(\sigma_a^{(\lambda)}(x)\right)'$ *is strictly monotone decreasing.*

When we apply an arithmetic operation to pliant inequalities we need to make sure that the operation is meaningful, i.e. the pliant inequalities represent the same sides of the fuzzy numbers. The following criteria formulates this requirement.

Criteria 4.4 *If* $\sigma_{a_1}^{(\lambda_1)}, \sigma_{a_2}^{(\lambda_2)}, \ldots, \sigma_{a_n}^{(\lambda_n)}$ *are inputs to an* n*-ary fuzzy arithmetic operation then*

$$\operatorname{sgn}(\lambda_1) = \operatorname{sgn}(\lambda_2) = \ldots = \operatorname{sgn}(\lambda_n)$$

must always hold.

4.2 Addition

Theorem 4.5 *Addition is closed over pliant inequalities and the addition function can be given as*

$$\sigma_{a_1}^{(\lambda_1)} \oplus \ldots \oplus \sigma_{a_n}^{(\lambda_n)} = \sigma_{a}^{(\lambda\ )} \qquad n \geq 1$$

where

$$\frac{1}{\lambda} = \sum_{i=1}^{n} \frac{1}{\lambda_i} \; and \; a = \sum_{i=1}^{n} a_i .$$

Proof. We prove by induction, if $i = 1$ then the statement is trivially true. Now let us assume that it holds for $i = n-1$ and prove it for $i = n$,

$$\delta = \underbrace{\sigma_{a_1}^{(\lambda_1)} \oplus \ldots \oplus \sigma_{a_{n-1}}^{(\lambda_{n-1})}}_{\sigma_{a'}^{(\lambda')}} \oplus \sigma_{a_n}^{(\lambda_n)} = \sigma_{a'}^{(\lambda')} \oplus \sigma_{a_n}^{(\lambda_n)}$$

where

$$\frac{1}{\lambda'} = \sum_{i=1}^{n-1} \frac{1}{\lambda_i} \; and \; a' = \sum_{i=1}^{n-1} a_i .$$

Now by using Lemma 2.1 we have

$$\delta^{-1}(z) = \left(\sigma_{a'}^{(\lambda')}\right)^{-1}(z) + \left(\sigma_{a_n}^{(\lambda_n)}\right)^{-1}(z) =$$

$$= \frac{1}{-\lambda'} \ln\left(\frac{1-z}{z}\right) + a' + \frac{1}{-\lambda_n} \ln\left(\frac{1-z}{z}\right) + a_n =$$

$$= \left(\sum_{i=1}^{n-1} \frac{1}{-\lambda_i} + \frac{1}{-\lambda_n}\right) \ln\left(\frac{1-z}{z}\right) + \left(\sum_{i=1}^{n-1} a_i + a_n\right) =$$

$$= \frac{1}{-\lambda} \ln\left(\frac{1-z}{z}\right) + a. \tag{2}$$

If $\sum_{i=1}^{n} \frac{1}{\lambda_i} \neq 0$ then $\delta^{-1}(z)$ is a strictly monotone function and inverse of a pliant inequality. Therefore $\delta(x)$ is a pliant inequality with λ and a parameters:

$$\delta(x) = \left(\sigma_{a_1}^{(\lambda_1)} \oplus \ldots \oplus \sigma_{a_n}^{(\lambda_n)}\right)(x) = \frac{1}{1+e^{-\lambda}(x-(a))} = \sigma_a^{(\lambda)}(x). \tag{3}$$

If $\sum_{i=1}^{n} \frac{1}{\lambda_i} = 0$ then the addition function does not exist since $\delta^{-1}(z) = a$ is a constant thus has no inverse. □

4.3 Multiplication by Scalar

Theorem 4.6 *Let given* $\sigma_a^{(\lambda)}(x)$ *sigmoid function.*

The scalar multiplication of the sigmoid function is:

$$c \odot \sigma_a^{(\lambda)}(x) = \sigma_{a'}^{(\lambda')}(x)$$

where

$$\lambda' = \frac{\lambda}{c} \qquad a' = ca$$

Proof. Using Lemma 2.1 gives us

$$\left(\sigma_a^{(\lambda)}(x)\right)^{-1} = c\left(-\frac{1}{\lambda}\ln\left(\frac{1-x}{x}\right) + a\right)$$

From here we have

$$\sigma_{a'}^{(\lambda')}(x) = \frac{1}{1+e^{-\frac{\lambda}{c}(x-ca)}} = \frac{1}{1+e^{-\lambda'(x-a')}}$$

□

4.4 Subtraction

We can derive subtraction from addition and negation.

Lemma 4.7 *Negation is closed over pliant inequalities and the negation function can be given as*

$$\ominus \sigma_a^{(\lambda)} = \sigma_{-a}^{(-\lambda)}$$

Proof. Let

$$\delta = \ominus \sigma_a^{(\lambda)}$$

by using Lemma 2.1 we have

$$\delta^{-1}(z) = -\left(\left(\sigma_a^{(\lambda)}\right)^{-1}(z)\right) =$$

$$= \frac{1}{\lambda}\ln\left(\frac{1-z}{z}\right) - a = -\frac{1}{-\lambda}\ln\left(\frac{1-z}{z}\right) + (-a).$$

Therefore

$$\delta(x) = \ominus \sigma_a^{(\lambda)}(x) = \frac{1}{1+e^{-(-\lambda)(x-(-a))}} = \sigma_{-a}^{(-\lambda)}(x). \tag{4}$$

□

Theorem 4.8 *Subtraction is closed over pliant inequalities and the subtraction function can be given as*

$$\sigma_{a_1}^{(\lambda_1)} \ominus \sigma_{a_2}^{(\lambda_2)} = \sigma_{a_1-a_2}^{\left(\frac{1}{\frac{1}{\lambda_1}-\frac{1}{\lambda_2}}\right)}.$$

Proof. Let

$$\delta = \sigma_{a_1}^{(\lambda_1)} \ominus \sigma_{a_2}^{(\lambda_2)}$$

by using Lemma 2.1 and Lemma 4.7 we have

$$\delta^{-1} = \left(\sigma_{a_1}^{(\lambda_1)}\right)^{-1} - \left(\sigma_{a_2}^{(\lambda_2)}\right)^{-1} =$$

$$= \left(\sigma_{a_1}^{(\lambda_1)}\right)^{-1} + \left(-\left(\sigma_{a_2}^{(\lambda_2)}\right)^{-1}\right) =$$

$$= \left(\sigma_{a_1}^{(\lambda_1)}\right)^{-1} + \left(\ominus \sigma_{a_2}^{(\lambda_2)}\right)^{-1} =$$

$$= \left(\sigma_{a_1}^{(\lambda_1)} \oplus \left(\ominus \sigma_{a_2}^{(\lambda_2)}\right)\right)^{-1},$$

therefore

$$\left(\sigma_{a_1}^{(\lambda_1)} \ominus \sigma_{a_2}^{(\lambda_2)}\right) = \sigma_{a_1}^{(\lambda_1)} \oplus \left(\ominus \sigma_{a_2}^{(\lambda_2)}\right) \tag{5}$$

Substituting Eq. 4 and Eq. 3 into Eq. 5 we get the desired result

$$\left(\sigma_{a_1}^{(\lambda_1)} \ominus \sigma_{a_2}^{(\lambda_2)}\right) = \sigma_{a_1}^{(\lambda_1)} \oplus \sigma_{-a_2}^{(-\lambda_2)} = \sigma_{a_1-a_2}^{\left(\frac{1}{\lambda_1}-\frac{1}{\lambda_2}\right)}. \tag{6}$$

□

Note: The function does not exist in case of $\lambda_1 - \lambda_2 = 0$.

4.5 Multiplication and the n^{th} Power

Now let $\delta = \sigma_{a_1}^{(\lambda_1)} \otimes \sigma_{a_2}^{(\lambda_2)}$. By using Lemma 2.1 we get

$$\delta^{-1}(z) = \left[\left(\sigma_{a_1}^{(\lambda_1)}\right)^{-1}(z)\right]\left[\left(\sigma_{a_2}^{(\lambda_2)}\right)^{-1}(z)\right] =$$

$$= \left[\frac{1}{-\lambda_1}\ln\left(\frac{1-z}{z}\right) + a_1\right]\left[\frac{1}{-\lambda_2}\ln\left(\frac{1-z}{z}\right) + a_2\right] =$$

$$= \frac{1}{\lambda_1\lambda_2}\ln^2\left(\frac{1-z}{z}\right) + \left(\frac{a_2}{-\lambda_1} + \frac{a_1}{-\lambda_2}\right)\ln\left(\frac{1-z}{z}\right) + a_1a_2$$

$$= \frac{1}{\lambda_1\lambda_2}\ln^2\left(\frac{1-z}{z}\right) - \frac{\lambda_1a_1 + \lambda_2a_2}{\lambda_1\lambda_2}\ln\left(\frac{1-z}{z}\right) + a_1a_2 \tag{7}$$

Unfortunately this is not an inverse pliant inequality and it is not monotone. To obtain the roots of the function we set $\left(\sigma_{a_1}^{(\lambda_1)}\right)^{-1}(z) = 0$ to get $z_0 = \frac{1}{1+e^{\lambda_1 a_1}}$ and we set $\left(\sigma_{a_2}^{(\lambda_2)}\right)^{-1}(z) = 0$ to get $z_1 = \frac{1}{1+e^{\lambda_2 a_2}}$.

A complete analysis of δ^{-1} would require checking both the $\lambda_1\lambda_2 > 0$ and $\lambda_1\lambda_2 < 0$ cases. However by Criteria 4.4 we only need to examine the first case.

Let $\lambda_1\lambda_2 > 0$. In this case $\delta^{-1}(z) < 0$ when $z \in (min(z_0, z_1), max(z_0, z_1))$. By using the first derivative we get $z_{min} = \frac{1}{1+e^{\frac{1}{2}(\lambda_1 a_1 + \lambda_2 a_2)}}$. Let us transform Eq. 7 to get z on the left side of the equation

$$\frac{1}{\lambda_1\lambda_2}\ln^2\left(\frac{1-z}{z}\right)-\frac{\lambda_1 a_1+\lambda_2 a_2}{\lambda_1\lambda_2}\ln\left(\frac{1-z}{z}\right)+a_1a_2=x$$

$$\frac{1}{\lambda_1\lambda_2}\left(\ln\left(\frac{1-z}{z}\right)-\frac{1}{2}\left(\lambda_1a_1+\lambda_2a_2\right)\right)^2=x-a_1a_2+\frac{\left(\lambda_1a_1+\lambda_2a_2\right)^2}{4\lambda_1\lambda_2}$$

$$\left(\ln\left(\frac{1-z}{z}\right)-\frac{1}{2}\left(\lambda_1a_1+\lambda_2a_2\right)\right)^2=\lambda_1\lambda_2x+\frac{1}{4}\left(\lambda_1a_1-\lambda_2a_2\right)^2$$

$$\ln\left(\frac{1-z}{z}\right)-\frac{1}{2}\left(\lambda_1a_1+\lambda_2a_2\right)=\pm\sqrt{\lambda_1\lambda_2x+\frac{1}{4}\left(\lambda_1a_1-\lambda_2a_2\right)^2} \qquad (8)$$

We need to check two cases here. First, let $z\in(0,z_{min}]$. In this case $\delta^{-1}(z)$ is strictly monotone decreasing (thus has an inverse) and the left side of Eq. 8 is positive therefore

$$\ln\left(\frac{1-z}{z}\right)-\frac{1}{2}\left(\lambda_1a_1+\lambda_2a_2\right)=\sqrt{\lambda_1\lambda_2x+\frac{1}{4}\left(\lambda_1a_1-\lambda_2a_2\right)^2}.$$

From here we have

$$\ln\left(\frac{1-z}{z}\right)=\sqrt{\lambda_1\lambda_2x+\frac{1}{4}\left(\lambda_1a_1-\lambda_2a_2\right)^2}+\frac{1}{2}\left(\lambda_1a_1+\lambda_2a_2\right)$$

$$\frac{1-z}{z}=e^{\left(\sqrt{\lambda_1\lambda_2x+\frac{1}{4}\left(\lambda_1a_1-\lambda_2a_2\right)^2}+\frac{1}{2}\left(\lambda_1a_1+\lambda_2a_2\right)\right)}$$

$$z=\frac{1}{1+e^{\left(\sqrt{\lambda_1\lambda_2x+\frac{1}{4}\left(\lambda_1a_1-\lambda_2a_2\right)^2}+\frac{1}{2}\left(\lambda_1a_1+\lambda_2a_2\right)\right)}}$$

For $\left(\sigma_{a_1}^{(\lambda_1)}\otimes\sigma_{a_2}^{(\lambda_2)}\right)(x)$ this gives us

$$\delta'(x)=\left(\sigma_{a_1}^{(\lambda_1)}\otimes\sigma_{a_2}^{(\lambda_2)}\right)(x)=\frac{1}{1+e^{\left(\sqrt{\lambda_1\lambda_2x+\frac{1}{4}\left(\lambda_1a_1-\lambda_2a_2\right)^2}+\frac{1}{2}\left(\lambda_1a_1+\lambda_2a_2\right)\right)}}$$

which is not a pliant inequality though rather similar. The domain of $\delta'(x)$ is

$$x \in [-\frac{1}{\lambda_1 \lambda_2}\left(\frac{a_1\lambda_1 - a_2\lambda_2}{2}\right)^2, \infty).$$

For the second case we have $z \in [z_{min}, 1)$. Now $\delta^{-1}(z)$ is strictly monotone increasing (thus has an inverse) and the left side of Eq. 8 is negative. Now we have

$$\ln\left(\frac{1-z}{z}\right) = -\sqrt{\lambda_1\lambda_2 x + \frac{1}{4}(\lambda_1 a_1 - \lambda_2 a_2)^2} + \frac{1}{2}(\lambda_1 a_1 + \lambda_2 a_2)$$

$$\frac{1-z}{z} = e^{\left(-\sqrt{\lambda_1\lambda_2 x + \frac{1}{4}(\lambda_1 a_1 - \lambda_2 a_2)^2} + \frac{1}{2}(\lambda_1 a_1 + \lambda_2 a_2)\right)}$$

$$z = \frac{1}{1 + e^{\left(-\sqrt{\lambda_1\lambda_2 x + \frac{1}{4}(\lambda_1 a_1 - \lambda_2 a_2)^2} + \frac{1}{2}(\lambda_1 a_1 + \lambda_2 a_2)\right)}}$$

For $\left(\sigma_{a_1}^{(\lambda_1)} \otimes \sigma_{a_2}^{(\lambda_2)}\right)(x)$ we get

$$\delta''(x) = \left(\sigma_{a_1}^{(\lambda_1)} \otimes \sigma_{a_2}^{(\lambda_2)}\right)(x) = \frac{1}{1 + e^{\left(-\sqrt{\lambda_1\lambda_2 x + \frac{1}{4}(\lambda_1 a_1 - \lambda_2 a_2)^2} + \frac{1}{2}(\lambda_1 a_1 + \lambda_2 a_2)\right)}}$$

The domain of δ'' is $x \in [-\frac{1}{\lambda_1 \lambda_2}\left(\frac{a_1\lambda_1 - a_2\lambda_2}{2}\right)^2, \infty)$.

Note that here λ_i and a_i play *the analogous roles* of m_i and a_i in additive pliant multiplication.

Functions σ' and σ'' shall be used if we need to multiply pliant inequalities from the positive range of the x axis. If $\lambda_1, \lambda_2 < 0$ then σ' should be used because it maps the positive part of $\sigma_{a_1}^{(\lambda_1)}$ and $\sigma_{a_2}^{(\lambda_2)}$. If $\lambda_1, \lambda_2 > 0$ then σ'' should be used.

Fig. 7 demonstrates multiplication of $\sigma_{a_1}^{(\lambda_1)}$ and $\sigma_{a_2}^{(\lambda_2)}$ pliant inequalities. The parameters of the operands were $\lambda_1 = 0.9, a_1 = 4$ and $\lambda_2 = 0.8, a_2 = 7$ respectively. The result is shown as the dotted curve.

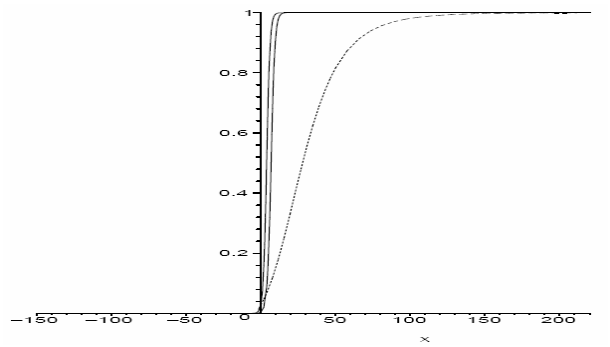


Figure 7
Multiplication of pliant inequalities

The calculations show that multiplication is not closed for pliant inequalities. Nevertheless we can approximate the result well with a pliant inequality. We construct this function to take the value $\frac{1}{2}$ at $a_1 a_2$ and let the tangent here be the same as of the multiplication function. For the tangent we have $\frac{1}{4}\frac{1}{\frac{a_1}{\lambda_2}+\frac{a_2}{\lambda_1}}$, and our approximation function is

$$\sigma_{a_1}^{(\lambda_1)} \otimes \sigma_{a_2}^{(\lambda_2)} \approx \sigma_{a_1 a_2}^{\left(\frac{1}{4}\frac{1}{\frac{a_1}{\lambda_2}+\frac{a_2}{\lambda_1}}\right)}.$$

Fig. 8 shows the approximation of the multiplication in Fig. 7. The approximating pliant inequality is plotted as the dotted curve.

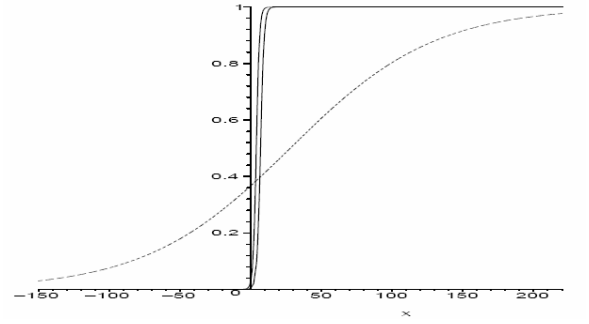


Figure 8
Approximation of pliant multiplication

We can use the approximation function for *large* values of a_1, a_2 or large values of $|\lambda_1|, |\lambda_2|$. In this case the function values are small for $x < 0$ thus the approximation is better.

Let us calculate the n^{th} power of $\sigma_a^{(\lambda)}$. This case

$$\delta^{-1}(z) = \left[\sigma_a^{(\lambda)}\right]^n = \left[\frac{1}{-\lambda}\ln\left(\frac{1-z}{z}\right) + a\right]^n$$

and then

$$\delta(x) = \frac{1}{1+e^{-\lambda\left(\sqrt[n]{x}-a\right)}}.$$

We can approximate the power function with a pliant inequality similarly to the multiplication function. Let the approximating power function take $\frac{1}{2}$ in a^n and let the tangent be the same here as of the power function. For the tangent this gives $\frac{1}{4}\frac{\lambda}{na^{n-1}}$ and our approximation function is

$$\left(\sigma_a^{(\lambda)}\right)^n \approx \sigma_{a^n}^{\left(\frac{1}{4}\frac{\lambda}{na^{n-1}}\right)}.$$

4.6 Division

Let $\delta = \sigma_{a_1}^{(\lambda_1)} \oslash \sigma_{a_2}^{(\lambda_2)}$. By using Lemma 2.1 we get

$$\delta^{-1}(z) = \frac{\left(\sigma_{a_1}^{(\lambda_1)}\right)^{-1}(z)}{\left(\sigma_{a_2}^{(\lambda_2)}\right)^{-1}(z)} = \frac{\frac{1}{-\lambda_1}\ln\left(\frac{1-z}{z}\right) + a_1}{\frac{1}{-\lambda_2}\ln\left(\frac{1-z}{z}\right) + a_2}. \tag{9}$$

Note that the function is undefined when the divisor equals zero, i.e. $z_0 = \frac{1}{1+e^{(\lambda_2 a_2)}}$. The function $\delta^{-1}(z)$ is strictly monotone decreasing in $(0, z_0)$ and strictly monotone increasing in $(z_0, 1)$. The limit of the function in

0 and in 1 is λ_2/λ_1. This means that the function does not take any value twice thus has an inverse and its domain is all real numbers except λ_2/λ_1. Let us transform Eq. 9 to get z on the left side of the equation

$$\frac{\frac{1}{-\lambda_1}\ln\left(\frac{1-z}{z}\right)+a_1}{\frac{1}{-\lambda_2}\ln\left(\frac{1-z}{z}\right)+a_2}=x$$

$$\frac{1}{-\lambda_1}\ln\left(\frac{1-z}{z}\right)+a_1=xa_2-\frac{1}{\lambda_2}\ln\left(\frac{1-z}{z}\right)x$$

$$\frac{1}{-\lambda_1}\ln\left(\frac{1-z}{z}\right)+\frac{x}{\lambda_2}\ln\left(\frac{1-z}{z}\right)=xa_2-a_1$$

$$\ln\left(\frac{1-z}{z}\right)=\frac{xa_2-a_1}{\frac{\lambda_2-x\lambda_1}{-\lambda_1\lambda_2}}=\frac{-\lambda_1\lambda_2}{\lambda_2-x\lambda_1}(xa_2-a_1)$$

$$\frac{1-z}{z}=e^{\frac{-\lambda_1\lambda_2}{\lambda_2-x\lambda_1}(xa_2-a_1)}$$

$$z=\frac{1}{1+e^{\frac{-\lambda_1\lambda_2}{\lambda_2-x\lambda_1}(xa_2-a_1)}}.$$

From here we have

$$\delta(x)=\left(\sigma_{a_1}^{(\lambda_1)}\oslash\sigma_{a_2}^{(\lambda_2)}\right)(x)=\frac{1}{1+e^{\frac{-\lambda_1\lambda_2}{\lambda_2-x\lambda_1}(xa_2-a_1)}}.$$

The result is not a pliant inequality. Fig. 9 presents the division of $\sigma_{a_1}^{(\lambda_1)}$ with $\sigma_{a_2}^{(\lambda_2)}$. The parameters were the same as in the multiplication example.

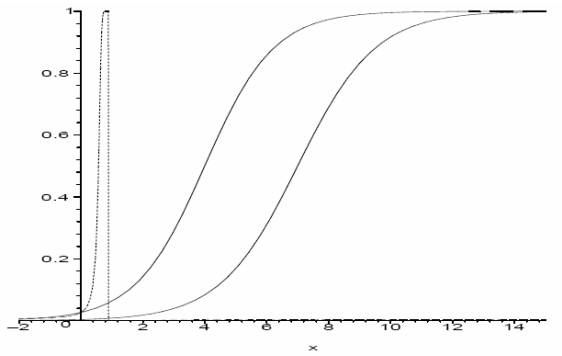


Figure 9
Division of pliant inequalities

Based on the approximation of the multiplication function we can use the following pliant inequality to approximate division.

$$\sigma_{a_1}^{(\lambda_1)} \oslash \sigma_{a_2}^{(\lambda_2)} \approx \sigma_{\frac{a_1}{a_2}}^{\frac{a_2}{4\lambda_1 - \frac{a_1}{a_2\lambda_2}}}.$$

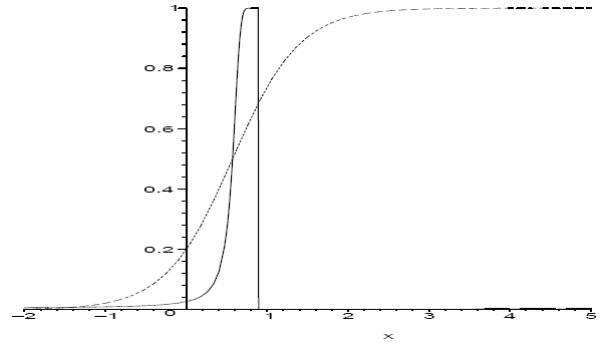


Figure 10
Approximation of pliant division

4.7 Properties of Operations

Theorem 4.9 *Addition is commutative and associative over pliant inequalities.*

Proof. These properties can be easily seen from the construction of $\frac{1}{m}$ and a in Theorem 4.5. □

Theorem 2.2 guarantees that multiplication over pliant inequalities is commutative, associative and distributive over addition. Now we prove these properties for the approximation of multiplication.

Theorem 4.10 *The approximation of multiplication over pliant inequalities is commutative, associative and distributive over addition.*

Proof. Commutativity

$$\sigma_{a_1}^{(\lambda_1)} \otimes \sigma_{a_2}^{(\lambda_2)} = \sigma_{a_1+a_2}^{\frac{\lambda_1\lambda_2}{a_1\lambda_1+a_2\lambda_2}} = \sigma_{a_2+a_1}^{\frac{\lambda_2\lambda_1}{a_2\lambda_2+a_1\lambda_1}} = \sigma_{a_2}^{(\lambda_2)} \otimes \sigma_{a_1}^{(\lambda_1)}$$

Associativity

Since

$$\frac{\frac{\lambda_1\lambda_2}{a_1\lambda_1+a_2\lambda_2}\lambda_3}{a_1a_2\frac{\lambda_1\lambda_2}{a_1\lambda_1+a_2\lambda_2}+a_3\lambda_3} = \frac{\frac{\lambda_1\lambda_2\lambda_3}{a_1\lambda_1+a_2\lambda_2}}{\frac{a_1a_2\lambda_1\lambda_2+a_1a_3\lambda_1\lambda_3+a_2a_3\lambda_2\lambda_3}{a_1\lambda_1+a_2\lambda_2}}$$

$$= \frac{\lambda_1\lambda_2\lambda_3}{a_1a_2\lambda_1\lambda_2+a_1a_3\lambda_1\lambda_3+a_2a_3\lambda_2\lambda_3} =$$

$$= \frac{\frac{\lambda_1\lambda_2\lambda_3}{a_2\lambda_2+a_3\lambda_3}}{\frac{a_1a_2\lambda_1\lambda_2+a_1a_3\lambda_1\lambda_3+a_2a_3\lambda_2\lambda_3}{a_2\lambda_2+a_3\lambda_3}} =$$

$$= \frac{\lambda_1\frac{\lambda_2\lambda_3}{a_2\lambda_2+a_3\lambda_3}}{a_1\lambda_1+a_2a_3\frac{\lambda_2\lambda_3}{a_2\lambda_2+a_3\lambda_3}},$$

we have

$$(\sigma_{a_1}^{(\lambda_1)} \otimes \sigma_{a_2}^{(\lambda_2)}) \otimes \sigma_{a_3}^{(\lambda_3)} = \sigma_{a_1a_2}^{\frac{\lambda_1\lambda_2}{a_1\lambda_1+a_2\lambda_2}} \otimes \sigma_{a_3}^{(\lambda_3)} =$$

$$= \sigma_{(a_1a_2)a_3}^{\left(\frac{\lambda_1\lambda_2}{a_1\lambda_1+a_2\lambda_2}\lambda_3\right)/\left(a_1a_2\frac{\lambda_1\lambda_2}{a_1\lambda_1+a_2\lambda_2}+a_3\lambda_3\right)} =$$

$$= \sigma_{a_1(a_2a_3)}^{\left(\lambda_1\frac{\lambda_2\lambda_3}{a_2\lambda_2+a_3\lambda_3}\right)/\left(a_1\lambda_1+a_2a_3\frac{\lambda_2\lambda_3}{a_2\lambda_2+a_3\lambda_3}\right)} =$$

$$= \sigma_{a_1}^{(\lambda_1)} \otimes \sigma_{a_2a_3}^{\frac{\lambda_2\lambda_3}{a_2\lambda_2+a_3\lambda_3}} = \sigma_{a_1}^{(\lambda_1)} \otimes (\sigma_{a_2}^{(\lambda_2)} \otimes \sigma_{a_3}^{(\lambda_3)}).$$

Distributivity

We shall use the following equality

$$\frac{\lambda_1\frac{\lambda_2\lambda_3}{\lambda_2+\lambda_3}}{a_1\lambda_1+(a_2+a_3)\frac{\lambda_2\lambda_3}{\lambda_2+\lambda_3}} = \frac{\frac{1}{\lambda_2+\lambda_3}}{\frac{1}{\lambda_2+\lambda_3}} \frac{\lambda_1\lambda_2\lambda_3}{a_1\lambda_1(\lambda_2+\lambda_3)+(a_2+a_3)\lambda_2\lambda_3} =$$

$$= \frac{\lambda_1\lambda_2\lambda_1\lambda_3}{a_1\lambda_1^2\lambda_2+a_1\lambda_1^2\lambda_3+a_2\lambda_1\lambda_2\lambda_3+a_3\lambda_1\lambda_2\lambda_3} =$$

$$= \frac{\frac{1}{(a_1\lambda_1+a_2\lambda_2)}\frac{1}{(a_1\lambda_1+a_3\lambda_3)}}{\frac{1}{(a_1\lambda_1+a_2\lambda_2)}\frac{1}{(a_1\lambda_1+a_3\lambda_3)}} \times$$

$$\times \frac{\lambda_1\lambda_2\lambda_1\lambda_3}{((a_1\lambda_1+a_3\lambda_3)\lambda_1\lambda_2+(a_1\lambda_1+a_2\lambda_2)\lambda_1\lambda_3)} =$$

$$= \frac{\frac{\lambda_1\lambda_2}{a_1\lambda_1+a_2\lambda_2}\frac{\lambda_1\lambda_3}{a_1\lambda_1+a_3\lambda_3}}{\frac{\lambda_1\lambda_2}{a_1\lambda_1+a_2\lambda_2}+\frac{\lambda_1\lambda_3}{a_1\lambda_1+a_3\lambda_3}}.$$

From here we have

$$\sigma_{a_1}^{(\lambda_1)} \otimes (\sigma_{a_2}^{(\lambda_2)} \oplus \sigma_{a_3}^{(\lambda_3)}) = \sigma_{a_1}^{(\lambda_1)} \otimes \sigma_{a_2+a_3}^{\frac{\lambda_2\lambda_3}{\lambda_2+\lambda_3}} = \sigma_{a_1(a_2+a_3)}^{\frac{\lambda_1\frac{\lambda_2\lambda_3}{\lambda_2+\lambda_3}}{a_1\lambda_1+(a_2+a_3)\frac{\lambda_2\lambda_3}{\lambda_2+\lambda_3}}} =$$

$$= \sigma_{a_1a_2+a_1a_3}^{\frac{\frac{\lambda_1\lambda_2}{a_1\lambda_1+a_2\lambda_2}\frac{\lambda_1\lambda_3}{a_1\lambda_1+a_3\lambda_3}}{\frac{\lambda_1\lambda_2}{a_1\lambda_1+a_2\lambda_2}+\frac{\lambda_1\lambda_3}{a_1\lambda_1+a_3\lambda_3}}} =$$

$$= \sigma_{a_1+a_2}^{\frac{\lambda_1\lambda_2}{a_1\lambda_1+a_2\lambda_2}} \oplus \sigma_{a_1+a_3}^{\frac{\lambda_1\lambda_3}{a_1\lambda_1+a_3\lambda_3}} =$$

$$= \left(\sigma_{a_1}^{(\lambda_1)} \otimes \sigma_{a_2}^{(\lambda_2)}\right) \oplus \left(\sigma_{a_1}^{(\lambda_1)} \otimes \sigma_{a_3}^{(\lambda_3)}\right).$$

□

5 Pliant Numbers and Operator Families

Pliant numbers can be decomposed to left hand side and right hand side fuzzy inequalities. In the previous sections we have used this decomposition to investigate two classes of suitable functions and their related arithmetic operations. After we have calculated the result of an arithmetic operation we need to combine the left hand side and the right hand side results using a fuzzy conjunction operator.

In this section we examine some classic fuzzy conjunction operators to investigate which ones are the most suitable for lines and pliant inequalities.

5.1 Pliant Numbers and Linear Function

Let us assume we would like to calculate the sum of two pliant numbers represented with lines. To this end we add the left lines and the right lines separately. Before applying conjunction we need to cut the results so they map to the $[0,1]$ interval. This can be achieved using the cut function defined as

$$[x] = \min(\max(x,0),1).$$

Fig. 11 presents a sample left and right line with the cut function applied.

The final step of addition is to construct the pliant number using a conjunction operator. Two cases should be considered here depending on how the two left and right hand side functions overlap.

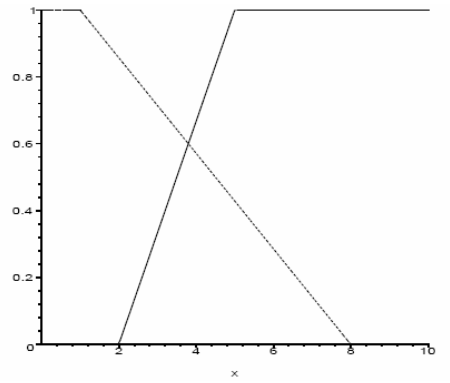


Figure 11
Cut function applied to lines

If the two functions do not intersect each other before they reach one then any conjunction operator can be used since any $c(x, y)$ conjunction satisfies $c(1, x) = x$ and $c(x,1) = x$ which guarantees that the lines remain intact.

Now if the two functions intersect each other before they reach one then we need to make sure that the conjunction operator does not distort the lines. Here we can also distinguish two cases depending whether the left line and the right line has the same absolute tangent or not. The left side of Fig. 12 shows a left line with $m = \frac{1}{4}, a = 4$ and a right line with $m = -\frac{1}{4}, a = 6$ parameters. The right side of the figure shows the same left line, but a different right line with parameters $m = -\frac{1}{6}, a = 6$.

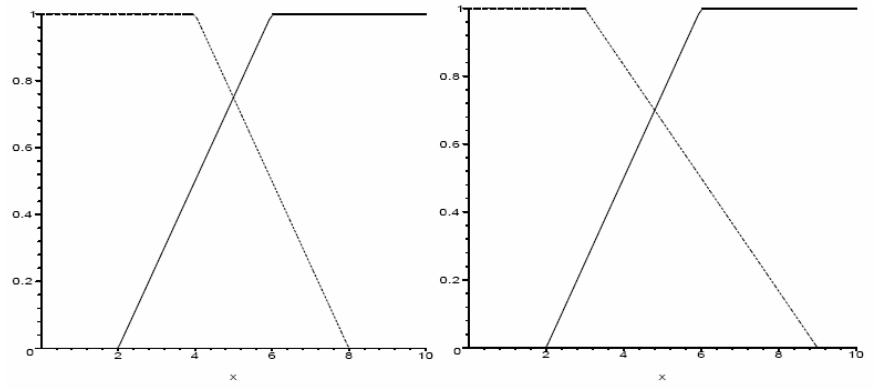


Figure 12
Lines with same and different absolute tangents

Let us now create a pliant number from these two linesets using the algebraic product $(a \cdot b)$ as shown in Fig. 13.

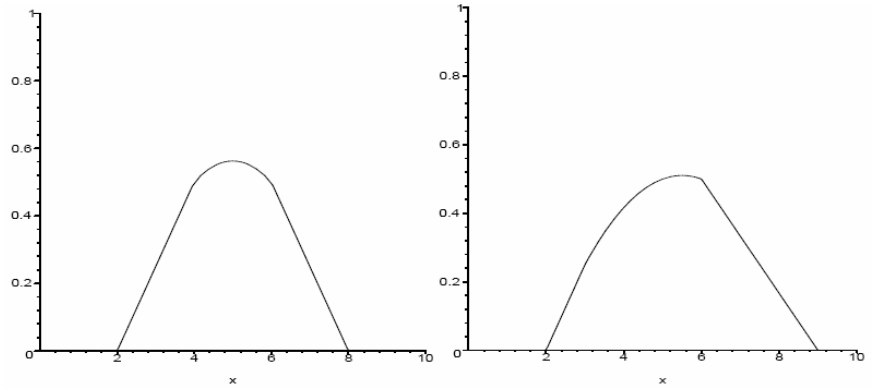


Figure 13
Product conjunction with same and different absolute tangent lines

As it can be seen from the figure the algebraic product conjunction distorts the lines around the point of their intersection.

Let us now apply the classic minimum conjunction.

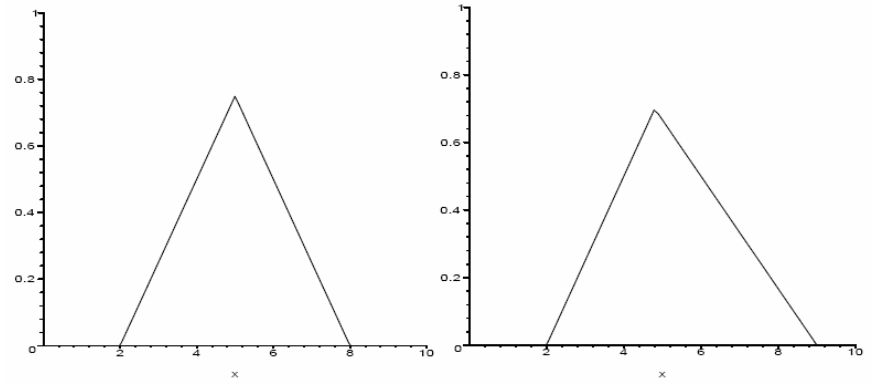


Figure 14
Minimum conjunction with same and different absolute tangent lines

Fig. 14 shows that the minimum conjunction simply cuts everything above the intersection point but leaves the remaining line segments undistorted.

Let us see what happens if we use the Lukasiewicz or bounded operator. $([x+y-1])$.

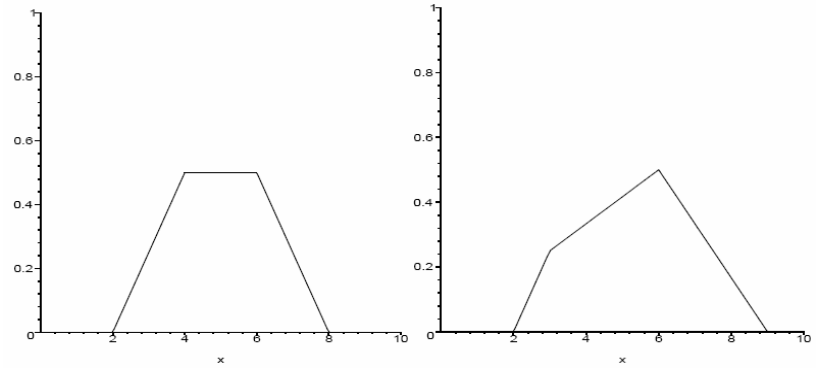


Figure 15
Bounded conjunction with same and different absolute tangent lines

As it can be seen from Fig. 15 the results are different from each other. On the left side of the figure the lines had the same absolute tangent which resulted in a trapezoid form while on the right side the lines had different absolute tangents which resulted in a cut-off point on the line with a greater absolute tangent value.

The results from the minimum and the Lukasiewicz conjunction resulted in lines which makes them a good choice for creating pliant numbers from additive pliant.

5.2 Pliant Numbers with Pliant Inequalities

Let us assume we would like to calculate the sum of two pliant numbers represented with pliant inequalities. This case we need to add two left hand side pliant inequalities together and add two right hand side pliant inequalities together. We can depict the operation using the following notation.

$$\begin{bmatrix} \lambda_l^1 & \lambda_r^1 \\ a_l^1 & a_r^1 \end{bmatrix} \oplus \begin{bmatrix} \lambda_l^2 & \lambda_r^2 \\ a_l^2 & a_r^2 \end{bmatrix} = \begin{bmatrix} \lambda_l & \lambda_r \\ a_l & a_r \end{bmatrix}$$

where

$$\frac{1}{\lambda_l} = \frac{1}{\lambda_l^1} + \frac{1}{\lambda_l^2}, \qquad \frac{1}{\lambda_r} = \frac{1}{\lambda_r^1} + \frac{1}{\lambda_r^2} \text{ and } a_l = a_l^1 + a_l^2, \qquad a_r = a_r^1 + a_r^2.$$

The subscripts denote the left and the right hand side operands respectively, superscripts denote the two pliant numbers and the results are denoted without superscripts. Theorem 4.5 defines the relationship between the parameters.

The final step of the addition is to construct the pliant number using a conjunction. Two cases should be considered here depending on how the two left and right hand side functions overlap.

First if the two functions do not intersect each other before they approach one then any conjunction operator can be used since any $c(x, y)$ conjunction satisfies $c(1, x) = x$ and $c(x,1) = x$ which guarantees that the shape of the curves will not be distorted. Fig. 16 presents this situation.

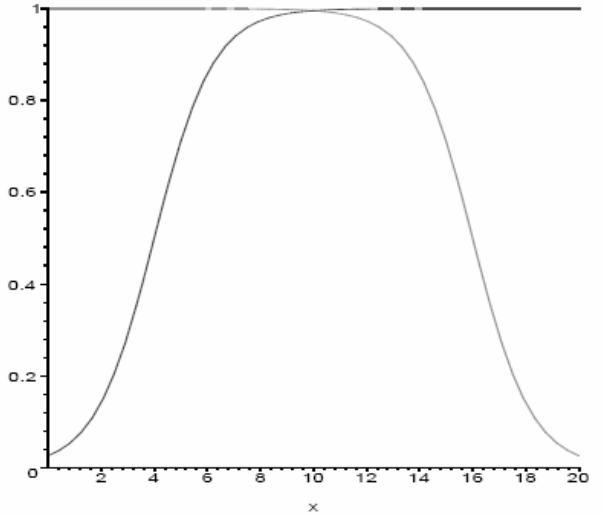


Figure 16
Two pliant inequalities

Now if the two functions intersect each other before they approach one then we need to make sure that the conjunction operator does not distort the shape of the curves. This is shown in the left side of Fig. 17.

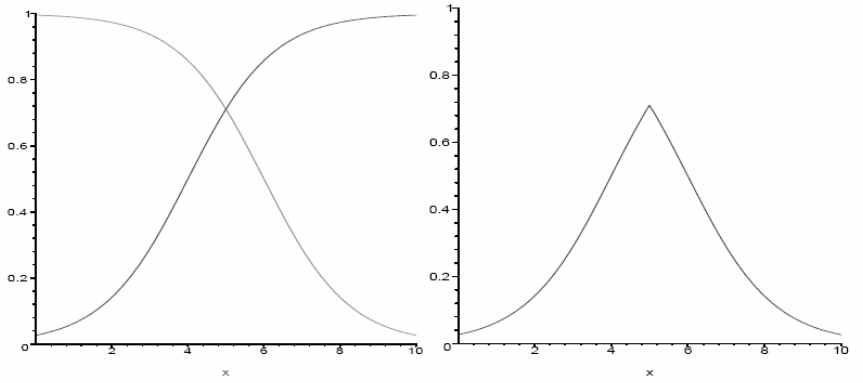


Figure 17
Two pliant inequalities, conjunction with $min(x, y)$

The right side of Fig. 17 shows the result of using the $min(x, y)$ conjunction. As it can be seen from the figure this operator cuts the functions sharply. Fig. 18 shows the result of conjunctions using the algebraic operator ($a \cdot b$) and the Lukasiewicz operator ($[x + y - 1]$).

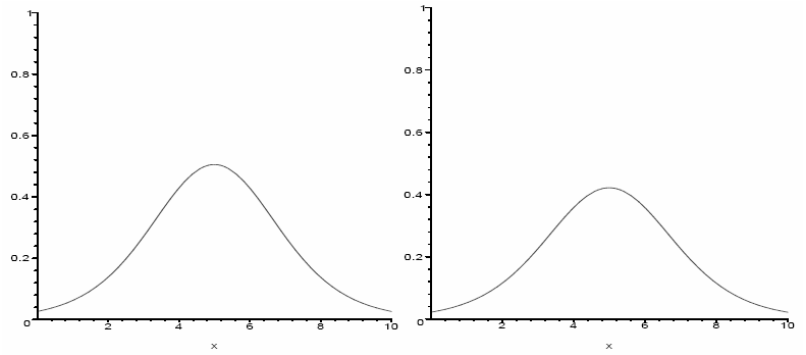


Figure 18
Conjunction with algebraic product and Lukasiewicz operator

While these conjunctions result in a smooth curve, they squash the operands along the y -axis. The averaging Dombi operator is given as

$$\bar{c}(x, y) = \frac{1}{1 + \frac{1}{2}\left(\frac{1-x}{x} + \frac{1-y}{y}\right)}.$$

Let us now use the averaging Dombi operator to construct the pliant number.

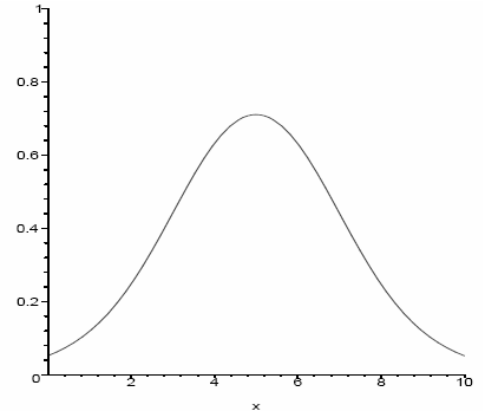


Figure 19
Conjunction with the Dombi operator

This case the result is a smooth curve that reaches its maximum value at the intersection of the curves. The averaging Dombi operator also retains the parameter values of the two functions as the following calculations show.

$$\bar{c}\left(\sigma_{a_l}^{(\lambda_l)}, \sigma_{a_r}^{(\lambda_r)}\right) = \frac{1}{1+\frac{1}{2}\left(\frac{1-\sigma_{a_l}^{(\lambda_l)}}{\sigma_{a_l}^{(\lambda_l)}} + \frac{1-\sigma_{a_r}^{(\lambda_r)}}{\sigma_{a_r}^{(\lambda_r)}}\right)} =$$

$$= \frac{1}{1+\frac{1}{2}\left(e^{-\lambda_l(x-a_l)} + e^{-\lambda_r(x-a_r)}\right)}.$$

This enables us to easily decompose the result for further arithmetic operations. These properties makes the Dombi operator a good choice for constructing pliant numbers based on pliant inequalities.

Acknowledgements

This research was partially supported by the TÁMOP-4.2.2/08/1/2008-0008 program of the Hungarian National Development Agency.

References

[1] D. Dubois and H. Prade. Operations on Fuzzy Numbers. *Int. J. Systems Science*, 9:613-626, 1978

[2] D. Dubois and H. Prade. Fuzzy Members: An Overview. *Analysis of Fuzzy Information*, Vol. I., CRC Press, Boca Raton, FL, 3-39, 1987

[3] D. Dubois and H. Prade. Special Issue on Fuzzy Numbers. *Fuzzy Sets and System*, 24(3), 1987

[4] R. Fullér and R. Mesiar. Special Issue on Fuzzy Arithmetic. *Fuzzy Sets and System*, 91(2), 1997

[5] R. Jain. Tolerance Analysis Using Fuzzy Sets. *Int. J. System Science*, 7: 1393-1401, 1976

[6] A. Kaufmann and M. M. Gupta. *Introduction to Fuzzy Arithmetic-Theoryand Applications.* Van Nostrand Reinhold, New York, 1985

[7] A. Kaufmann and M. M. Gupta. *Fuzzy MAthematical Models in Engineering and Management Science.* North-Holland, Amsterdam, 1988

[8] M. Mares. *Computation Over Fuzzy Quantities.* CRC Press, Boca Raton, FL, 1994

[9] M. Mizumoto and K. Tanaka. The Four Operations of Arithmetic on Fuzzy Numbers. *Syst. Comput. Controls*, 7(5): 73-81, 1976

[10] M. Mizumoto and K. Tanaka. Algebraic Properties of Fuzzy Numbers. *Proc. Int. Conf. On Cybernetics and Society*, Washington, DC, 559-563, 1976

[11] S. Nahmias. Fuzy Variables. *Fuzzy Sets and System*, 1:97-110, 1978

[12] H. T. Nguyen. A Note on the Extension Principle for Fuzzy Sets. *J. Math. Anal. Appl.*, 64:369-380, 1978

[13] R. C. Young. The Algebra of Many-valued Quantities. *Math. Ann.*, 104:260-290, 1931

[14] L. A. Zadeh. The Concept of a Linquistic Variable and its Application to Approximate Reasoning. *Information Sciences*, Part I:8, 199-249, Part II:8, 301-357, Part III:9, 43-80, 1975

Dynamic Model Identification for Industrial Robots

Ngoc Dung Vuong, Marcelo H. Ang Jr.

Department of Mechanical Engineering, Engineering Faculty
National University of Singapore
9 Engineering Dr 1, 117576 Singapore
<ndvuong, mpeangh>@nus.edu.sg

Abstract: In this paper, a systematic procedure for identifying the dynamics of industrial robots is presented. Since joint friction can be highly nonlinearwith time varying characteristics in the low speed region,a simple and yet effective scheme has been used to identify the boundary velocity that separates this "dynamic" friction region from its static region. The robot's dynamic model is then identified in this static region, where the nonlinnear friction model is reduced to the linear-in-parameter form. To overcome the drawbacks of the least squares estimator, which does not take in any constraints, a nonlinear optimization problem is formulated to guarantee the physical feasibility of the identified parameters. The proposed procedure has been demonstrated on the first four links of the Mitsubishi PA10 manipulator, an improved dynamic model was obtained and the the effectiveness of the proposed identification procedure is demonstrated.

Keywords: Dynamic Modeling, Model Identification, Friction Models, Model-based Control

1 Introduction

The robot's dynamic model is required in the implementation of most advanced model-based control schemes. The dynamic model is crucial because it can be used to linearize the nonlinear system in both joint space [1] and task space [2]. Since the robot's dynamic parameters are normally not available for industrial manipulators, proper procedures should be carried out to identify these parameters.

One way to identify the dynamic parameters is to dismantle the robot and measure link by link [3]. However, it is obvious that this approach is not always feasible in practice. Another problem with the dismantling approach is that it does not account for the effects of joint friction.

In order to account for joint frictions, several methods were proposed. These methods can be roughly divided into two groups: to identify joint friction and rigid body dynamics separately [4] or to identify joint friction and rigid body dynamics simultaneously [5-7].The former first identifies the friction parameters for each joint and then continues to identify the rigid body dynamic parameters using the identified friction parameters. Since friction parameters are identified joint by joint, nonlinear dynamic friction models such as Stribeck and/or hysteresis effects can be considered [8].The main drawback of this method comes from the fact that friction can be much time-varying [7]. Moreover, friction forces/torques are always coupled to the inertial forces/torques, thus, one cannot be precisely identified without the other. It is also argued that it is more tedious to identify friction parameters and rigid body dynamic parameters separately.

From the literature, more researchers adopt the latter method, i.e. to identify joint frictions and the rigid body dynamics at the same time [5-7]. It is worth noting that the robot's dynamic model (excluded joint frictions) can be linearized w.r.t to its parameters. Thus, many proposed identification methods was accomplished based on the assumption that joint frictions can be modeled in a linear-in-parameter form. However, this linearity is not valid for velocities. At slow velocities, the friction parameters exhibit some dynamics, and we refer to this region as the "dynamic" region of friction. When velocities exceed a threshold velocity, the friction parameters become "static" and the friction is now linear in the parameter form. We therefore refer to this region as the "static or linear" regiion. The use of the linear friction model outside this linear region can lead to significant errors on the identified parameters as demonstrated in [9]. In this paper, a simple and effective scheme which has been introduced in our previous work [9] will be used to identify the threhshold velocity that separates the joint friction into dyanimic (and nonlinear) and static (and linear in parameter) regions regions. The robot dynamic model is then identitied only in the linear region, thus more accurate dynamic parameters are obtained.

Since the robot's dynamic model is linear w.r.t its parameters, these dynamic parameters can be identified using the well-known least-squares estimator. Note that not all ten inertial parameters of each robot's link can be identified due to the raltive configuration of the links of the robot. It is therefore necessary to reduce/simplify the robot model to ensure that the observation matrix of the least-square estimator has full rank [10]. This problem can be solved either symbolically [10] or numerically [11].

Since the measured torques are normally noisier than the measured position, a proper trajectory should be designed to ensure the robustness of the identified results [12]. To guarantee the robustness of the estimation process, several criterions have been proposed in the literature such as maximizing the determinant or minimizing the condition number of the observation matrix [6]. Note that all the above criteria result in solving a nonlinear constrained optimization problem. The results from this optimization problem are the so-called exciting/optimal trajectory

that can guarantee the excitation of all the parameters to be identified. Because of the complexity of the dynamic model, genetic algorithm (GA) is used in this paper to find out the above optimal trajectory.

It is worth pointing out that the above exciting trajectory can only account for the uncertainties of the measured torque. In practice, uncertainties can also occur in the motion data (i.e. joint position, velocity and acceleration). Moreover, due to the fact that most industrial manipulators do not come with velocity and acceleration sensor, thus, these information are normally obtained through numerical differentiation of the joint position measurements. As a result, the quality of the observation matrix of the least square estimator can be significantly degraded. A direct consequence of this observation is that the results from the least square estimator can deviated from its true value. Since no constraints are imposed on the least-squares technique, it is possible for the least-squares estimator to produce results which are physically impossible [13-14]. Although there are other methods to cope with uncertainties on the observation matrix such as the maximum likelihood method [15], most of them do not consider the physical feasibilty of the identified parameters as an important criteria. Noting that a physically non-feasible dynamic model cannot be used in model-based control because this model can result in a non-possitive definite inertial matrix, thus, destabilize the closed loop control system. One promising solution for this problem is to use constrained optimization tools to adjust the least-squares result [16]. However, this method requires the initial guess of the virtual parameters which are not always available in practice.

Although there is a vast amount of results on the dynamic identification topics in the literature, a systematic procedure which includes all the above considerations is still missing. Thus, the aim of this paper is to present a systemantic procedure for identifying the robot's dynamic model. This dynamic model can then be used in advanced model-based controllers.

2 Rigid Body Modeling and Identification

2.1 Rigid Body Modeling

It is well known that the dynamic model of an n-degree-of-freedom (n-DOF) serial manipulator can be expressed in the following analytical form:

$$M(q)\ddot{q} + C(\dot{q},q) + G(q) + \Gamma_{fric} = \Gamma \tag{1}$$

where:

- $\ddot{q}, \dot{q}, q$ are n by 1 vectors of joint acceleration, velocity and position, respectively.

- $M(q), C(\dot{q}, q), G(q)$ are the inertial matrix, Coriolis-Centrifugal and gravity vector in joint space.

- Γ_{fric} is an n by 1 vector of joint friction and Γ is a n by 1 vector of force/torque at each joint.

For identification purpose, the above equation is re-written in the linear form:

$$W(\ddot{q}, \dot{q}, q, DH)h + \Gamma_{fric} = \Gamma \tag{2}$$

Here, DH is the kinematic parameters from the Denavit-Hartenberg parameters and h is a 10n x 1 vector of the inertial parameters:

$$h_i = \left[XX_i, XY_i, XZ_i, YY_i, YZ_i, ZZ_i, mX_i, mY_i, mZ_i, m_i \right]^T \tag{3}$$

$$h = \left[h_1 \quad ... \quad h_n \right]^T \tag{4}$$

where $\left(XX_i, XY_i, XZ_i, YY_i, YZ_i, ZZ_i \right)$ are the inertial tensor of link i, $\left(mX_i, mY_i, mZ_i, m_i \right)$ are the first moments and the link mass. Noting that, here, we only focus on the inertial parameters of the links. The rotor inertias of motors are assumed to be known because these values are normally available from the motor specs. From (2), it is clear that if joint friction model are linear w.r.t its parameters, the problem of identifying the dynamic model is a linear problem. In the next section, condition for which the linear-in-parameter friction model is valid will be derived.

2.2 Nonlinear Friction Model and Boundary Velocity

Although joint frictions is complicated in reality, simple model, which is the combination of viscous and Coulomb friction, is normally used to describe the friction phenomenon for all joint(s):

$$\Gamma_{fric\_i} = F_{ci} sign(\dot{q}_i) + F_{vi} \dot{q}_i \tag{5}$$

where where F_{ci} and F_{vi} are Coulomb and viscous friction coefficients of joint i respectively. However, this assumption can lead to a significant degradation of the accuracy of the identified parameters. One solution for this problem is to make use of the boundary velocity where the nonlinear and linear friction are separated as discussed in [9]. By analyzing the velocity-torque map, one should be able to identify the boundary velocity for each joint.

The following part briefly describes the step-by-step procedures for obtaining the:

- Step 1: re-mount the manipulator in such a way that the gravity has no effect on the joint of interest. Apply a sinusoidal torque to the joint. Notice that the frequency and magnitude of this signal have to be chosen in such a way that the result joint motion is within the joint limit and the motion also excites the dynamic friction. During this step, $(q, \dot{q}, \ddot{q}, \tau)_{i=1:N}$ is recorded (N is the number of recorded points). Since only one joint is excited at the time, the equation of motion of the system is:

$$I\ddot{q} + F_C sign(\dot{q}) + F_V \dot{q} = \tau \tag{6}$$

where I, F_C, F_V are the lumped inertia, Coulumb and viscous friction coefficients of the joint of interest. If (6) can describe the friction model at joints, I, F_C, F_V can be resolved from the following linear system:

$$\begin{bmatrix} \ddot{q}_1 & sign(\dot{q}_1) & \dot{q}_1 \\ \dots & \dots & \dots \\ \ddot{q}_M & sign(\dot{q}_M) & \dot{q}_M \end{bmatrix} \begin{bmatrix} I \\ F_C \\ F_V \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \dots \\ \tau_M \end{bmatrix} \tag{7}$$

where $M \leq N$ is the number of data points which are used for the identification.

- Step 2: slowly increasing $\dot{q}_{thres}$ from 0 to $\max(|\dot{q}_i|)$. Solve (7) for the parameters of the dynamic model using only the data points for which $|\dot{q}_i| > \dot{q}_{thres}$. The parameters of the dynamic model should be constant. By analyzing the convergence of the inertial parameter $\hat{I}$, one can experimentally find out the region in which the linear friction model is held. Based on this result, we can actually reconstruct the joint velocity vs. friction torque plot (or friction map). For instance, Figure 1a shows the convergence of $\hat{I}$ w.r.t $\dot{q}_{thres}$, Figure 1b shows the friction map of the first joint of the PA10 manipulator.

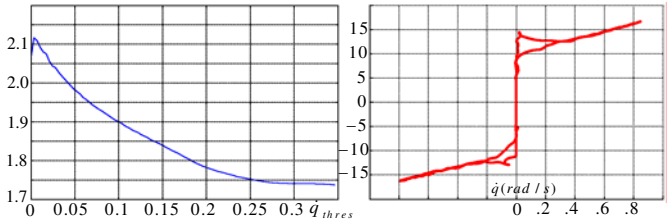


Figure 1

(a) $\dot{q}_{thres}$ vs. $\hat{I}$, (b) $\dot{q}$ vs. $\tau_{friction}$ of the first joint of the PA10 manipulator

- Step 3: The experiment is then repeated for the rest of the joints. The resulting $\dot{q}_{thres}$ will be used as constraints in designing the exciting trajectory (as presented in the next section).

In summary, if joint velocity is outside the range $(-\dot{q}_{thres}, \dot{q}_{thres})$, joint friction can be modeled as a combination of Coulomb and viscous friction (Eq. 5). By incorporating (2) and (5), the robot dynamic model can be rewritten as:

$$W(\ddot{q}, \dot{q}, q, DH)_c h_c = \Gamma \tag{8}$$

where W_c, h_c are the combinations of inertial parameters and friction coefficients:

$$W_c = \begin{bmatrix} W_{1,:} & sign(\dot{q}_1) & \dot{q}_1 & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... \\ W_{n,:} & 0 & 0 & ... & sign(\dot{q}_n) & \dot{q}_n \end{bmatrix}, h_c = \begin{bmatrix} h \\ F_{c1} \\ F_{v1} \\ ... \end{bmatrix} \tag{9}$$

It is worth noting that equation (9) indicates that in order to re-solve for h_c, W_c matrix has to be full rank. It is well-known that not all the inertial parameters contribute to the dynamic behaviour of the robot [1, 10, 17]; thus, a set of identifiable parameters (the so-called *base parameters* [10]) should be deduced from h. For instance, the original dynamic parameters of the 7-DOF Mitsubishi PA10 manipulator h has 70 parameters but the final identified dynamics of the manipulator is reduced into 18 lumped-parameters [14]. The final form of the dynamic model becomes:

$$W_b h_b = \Gamma \tag{10}$$

where h_b is comprised of the base parameters and linear friction model. Theoretically, by resolving (10), one can accurately estimate the inertial parameters h_b provided that the observation matrix W_b and the joint torque Γ can be accurately obtained. In practice, these assumptions are always violated. As a result, the identification experiment should be designed in such a way that the results from the least-square estimator are robust w.r.t to the noise. This observation leads to the discussion in the next section: the design of the exciting trajectory.

2.3 Exciting Trajectory

In order to estimate h_b from (10), $\{W(\ddot{q}, \dot{q}, q)_b, \Gamma_b\}_i$ need to be acquired through the identification experiment. By stacking the matrix together, the observation matrix can be formed as follows:

$$W_o = \begin{bmatrix} W_{b1} \\ ... \\ W_{bN} \end{bmatrix}, \Gamma_o = \begin{bmatrix} \Gamma_{b1} \\ ... \\ \Gamma_{bN} \end{bmatrix} \tag{11}$$

Theoretically, as long as the determinant of the observation matrix W_o, which depends on the exciting trajectory which has been used in the identification experiment, is non-zero, the unknown parameters h_b can be estimated by the well-known least-squares/weighted least-square estimator:

$$\hat{h}_b = \left(W_o{}^T W_o\right)^{-1} W_o{}^T \tau_o \tag{12a}$$

However, if the measured torques are corrupted by noise, a constraint should be imposed on the experiment trajectory to ensure the robustness of the identified results. Physically, finding this constraint is equivalent to finding an optimal trajectory that can excite most the identified parameters. Several criteria have been proposed in literature [18]. In this paper, minimizing the condition number and maximizing the smallest singular value of the observation matrix W_o as in [6] is adopted. Moreover, if the information on the noise is available, weighted least-square can be used:

$$\hat{h}_b = \left(W_o{}^T R W_o\right)^{-1} W_o{}^T R \tau_o \tag{12b}$$

where R is the inverse of the covariance noise matrix [19].

Notice that, because we want to minimize the effect of the non-linear friction on the identified result(s), only the data points which have velocities above a threshold/boundary value (from the previous section) are considered. This differs from other researchers which normally take into account all data points along the optimum trajectory. Since the optimum trajectory will be executed on the manipulator, parameterizing the optimum trajectory is also an important step. Two most common type(s) are the quintic polynomial trajectory [6] and periodic trajectory [20]. The former is suitable for most of industrial manipulator(s) which only accepts simple velocity command while the later targets the open-architecture controller which allows user to program an arbitrary trajectory. In this paper, periodic trajectory, which can be parameterized as a sum of finite Fourier series (13), is adopted because of their advantages in terms of signal processing [21]:

$$q_i(t) = q_{i0} + \sum_{l=1}^{N} a_{il} \sin(w_f lt) - b_{il} \cos(w_f lt) \tag{13a}$$

$$\dot{q}_i(t) = \sum_{l=1}^{N} a_{il} w_f l \cos(w_f lt) + b_{il} w_f l \sin(w_f lt) \tag{13b}$$

$$\ddot{q}_i(t) = \sum_{l=1}^{N} -a_{il}\left(w_f l\right)^2 \sin(w_f lt) + b_{il}\left(w_f l\right)^2 \cos(w_f lt) \tag{13c}$$

where w_f is the fundamental frequency of the excitation trajectories and should be carefully chosen not to excite the un-modeled dynamics of the manipulator. The problem of finding the optimal trajectory becomes determining the coefficients (q_{i0}, a_k^i, b_k^i) in order to minimize the following cost function:

$$f(q_i(t)) = \lambda_1 cond(W_c) + \lambda_2 \frac{1}{\sigma_0(W_c)} \tag{14}$$

where the scalar λ_1 and λ_2 represent the relative weights between the condition number of the observation matrix: $cond(W_c)$ and its minimum singular value: $\sigma_0(W_c)$ [6]. Note that the above problem is a constrained optimization problem because physical limits of joint position, velocity and acceleration have to be considered. As can be seen from (9) and (10), the cost function is nonlinear and discontinuous e.g. the *sign* function in (9). This can make the optimization process become significantly difficult. In practice, one can avoid the discontinuity by replacing the $sign(\dot{q}_i)$ function in (9) with an approximated continuous function such as $atan(c\dot{q}_i)$. The extra coefficient c is used to adjust the steepness of the slope when $\dot{q}$ approaches zero. Due to the complexity of the problem, a good initial guess for this optimization is hard to achieve. Thus, a genetic algorithm (GA) is used to solve the above optimization problem.

Once the optimization has been solved, the optimum trajectories $q_i(t)$ for all joints are obtained. The manipulator will be commanded to follow this optimal trajectory by any available controller. For instance, an independent joint control scheme which includes a high-gain PID controller at each joint was used in our experiment. The responses of the robot along the trajectories will be recorded. It is worth noting that the collected data should be pre-processed as suggested in [20] in order to improve the data quality before using them to estimate the dynamic parameter. A brief description is as follows:

- Firstly, the joint position data can be filtered by a low-pass filter with an appropriate cut-off frequency which depends on the choice of the fundamental frequency w_f in (13). This is reasonable because the frequency components in the optimal trajectory from previous section are already predefined in the design state.
- If joint velocity and acceleration are not available due to the lack of joint sensors, this information can be obtained through a numerical differentiation. However, since the exciting trajectory are designed in the form of (13), a linear least square fit (15) can be performed to estimate

the coefficients (q_{i0}, a_{il}, b_{il}) of the actual optimal trajectory (i.e. the actual motion of the robot) as suggested in [15]:

$$q_i(t) = \begin{bmatrix} q_i(t=0) \\ q_i(t=t_1) \\ \dots \\ q_i(t=T_f) \end{bmatrix} = \begin{bmatrix} 1 & \sin(w_f 0) & -\cos(w_f 0) & \dots & \dots \\ 1 & \sin(w_f t_1) & -\cos(w_f t_1) & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \sin(w_f T_f) & -\cos(w_f T_f) & \dots & \dots \end{bmatrix} \begin{bmatrix} q_{i0} \\ a_{i1} \\ b_{i1} \\ \dots \end{bmatrix} \quad (15)$$

As a result, joint velocity and acceleration can be obtained by substituting these coefficients into Equations (13b) and (13c).

Note that the above method should only be used if the independent joint control scheme is able to control the manipulator to closely follow the optimal trajectory. The reason is because the above approach totally ignores all the frequentcy components that are not in the form of (13) in the observation matrix (the left-hand side of (10)). However, the measured torques (the right-hand side of (10)) are indepedently filtered, thus, it is possible that the information on two-side of equation (10) is not consistent.

2.4 Parameter Estimation

Although the unknown inertial parameters can be estimated by a least-square technique as in (12), there will be a potential problem on the identified results, the so-called physical feasibility of the results [13]. One promising solution for this problem is to use constrained optimization tools to adjust the least-squares result [16]. By doing this, the physical meaning of the identified parameters can be guaranteed by imposing appropriate constraints on the estimator. The physically feasible characteristic is especially useful for advanced control because it implies that the mass matrix $M(q)$ in (1) is always positive definite.

Motivated by the idea of virtual parameters [13], a constrained optimization is used in order to find the unknown inertial parameters. The input vector X to the optimization problem is:

$$X = \begin{bmatrix} h_{70\times 1} & F_{c1} & F_{v1} & \dots & \dots \end{bmatrix}^T \quad (16)$$

where h is the standard dynamic parameters of links as in (3). Constraints on the parameters h will be imposed in order to make sure that the result will always physically feasible. Based on this input, the base parameters vector h_b is calculated. This base parameters vector is then compared to the least-squares solution $\hat{h}_b$ from (12). The cost function for this constrained optimization is constructed as:

$$CF = \min\left(\alpha_1 \left\| W_o X_b - \tau_o \right\| + \alpha_2 \left\| X_b - h_c \right\|\right) \tag{17}$$

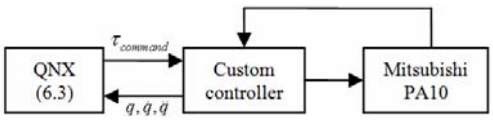
Here, the two scalars α_1, α_2 define how believable the least-squares solution is. Clearly the result of the above non-linear optimization problem will give us a set of physically feasible parameters which also minimizes the error between the measured and predicted torque.

Since the purpose of this paper is to demonstrate a step-by-step procedures for identifying the dynamic model of industrial robots, the above procedures is applied for the first four link of the PA10 manipulator. The results from the identification process has been verified by comparing the reconstructed torques and the measured torques for an arbitrary joint space trajectory. In addition, the identified model has been tested in a conventional computed-torque controller. A singificant improvement in terms of tracking errors was obtained which also shows the usefulness of the identified model.

3 Case Study – The Mitsubishi PA10 Manipulator

3.1 Experimental Testbed

To validate the proposed identification method, the identification procedures were applied to the first four links of the Mitsubishi PA10 manipulator. A custom controller has been used instead of the original controller in order to achieve a critical real-time performance. Seven custom amplifiers are installed together with an 8-axis motion controller card. A PC which is running the QNX 6.3 operating system is used to control the manipulator as depicted in Figure 2.



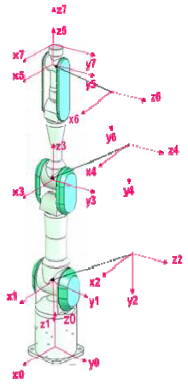


Figure 2
Experimental test bed

As mentioned above, the following steps were carried out in order to identify the dynamic model:

1 Derive the rigid dynamic model of the robot as in (9) and (10). Noting that the Coriolis-Centrifugal and gravity term is included in this model. Gravity

terms can only be set to zero when robot joints are considered separately as in Section 2.2.

2 Identify the boundary velocity $\dot{q}_{thres}$ in which the dynamic friction model becomes linear for each joint (see Table 1 for the experimental results).

Table 1
Boundary velocity for the first four links of the PA10

Joint	$\dot{q}_{thres}(rad/s)$
1	0.25
2	0.27
3	0.3
4	0.6

3 Carry out the optimum exciting trajectory as in Section 2.3. By making use of the Matlab Genetic Algorithm (GA) Toolbox, the optimum trajectory was found with the minimum condition number around 65.

4 Execute the optimum trajectory on the PA10; obtain the joint motion and joint torque data. Note that because the PA10 manipulator does not have joint torque sensor, the joint torques are obtained by measuring the motor currents. An independent joint control scheme is used at each joint to make the joints follow the reference/optimal trajectory.

5 The inertial parameters are estimated using the method as in Section 2.4. The equivalent virtual parameters as in (16) and (17) are shown in Table 2. Note that these parameters were obtained with the constraint $(m, I)_i > 0$ to make sure the physical feasibility of the identified results.

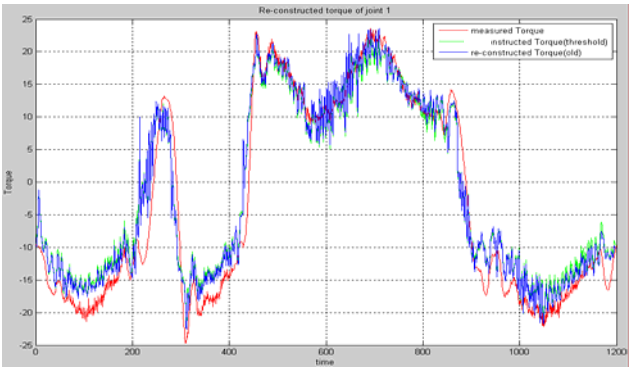
Table 2
Virtual parameters X (15) that minimize the cost function (16)

	1	2	3	4
XX	0.3811	0.0347	0.0236	0.0758
YY	0.6884	1.8651	0.0119	0.0073
ZZ	0.0356	3.006	0.1705	0.588
XY	0.9833	0.7599	0.5881	0.8631
XZ	1.1198	0.8079	0.5955	0.1214
YZ	0.317	0.0001	1.5338	0.019
xc	0.8353	2.3296	1.3338	0.8852
yc	0.8831	0.0827	0.4072	0.1501
zc	3.1872	0.4506	0.2006	1.6751
m	0.4696	0.2088	2.5769	3.7324

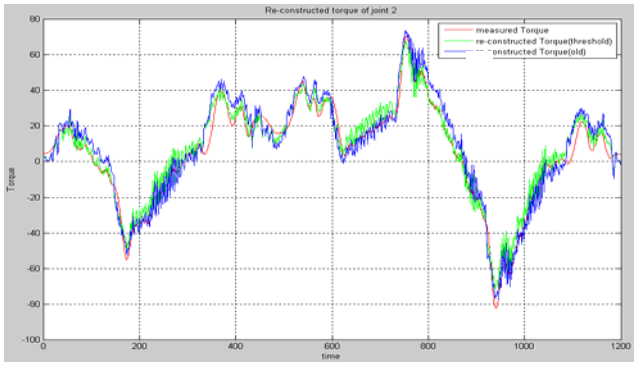
3.2 Model Verification

3.2.1 Torque Reconstruction

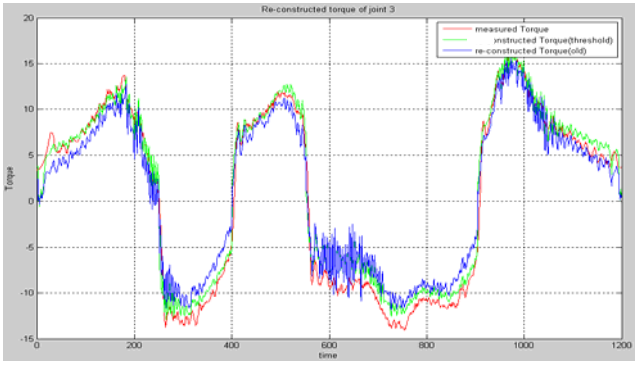
As noted in Section 2.2, the result of the above identification process is the parameter $\hat{h}_b$ which is the combination of the base parameters and joint friction coefficients. Since the base parameters are lump from the link inertias h, it is impossible to directly check the correctness of the identified parameters. Instead, the identified model is verified by comparing the reconstructed torques, which are generated from the identified model, and the measured torques, which are the actual joint torques that are used to control the manipulator. Since the major difference between the approach in this paper and others is the use of the boundary velocity, it is necessary to check whether the identified parameters using the boundary velocity has any advantages. To this end, two sets of data have been used to identify $\hat{h}_b$. The first set (set A) only includes the data points which have $|\dot{q}| > \dot{q}_{thres}$ while the second set (set B) includes all the experimental points. In the case of the PA10 manipulator, the number of data points in set A is about 30% of the number of data points in set B. Figure 3 shows the measured torques vs. the re-constructed torques of joint 1-4 for an arbitrary and different trajectory in joint space. Motion data $(\ddot{q}, \dot{q}, q)$ and joint torque (namely "measured torque") was recorded. The "re-constructed torque" is then computed as (10). In Figure 3, red represents the measured torque (MT), blue represents the re-constructed torque using $\hat{h}_b^B$ (RTall $= W_b \hat{h}_b^B$) and green represents the re-constructed torque using $\hat{h}_b^A$ (RTthres $= W_b \hat{h}_b^A$). Noting that the time scale for the x-axis is time 10 (ms).



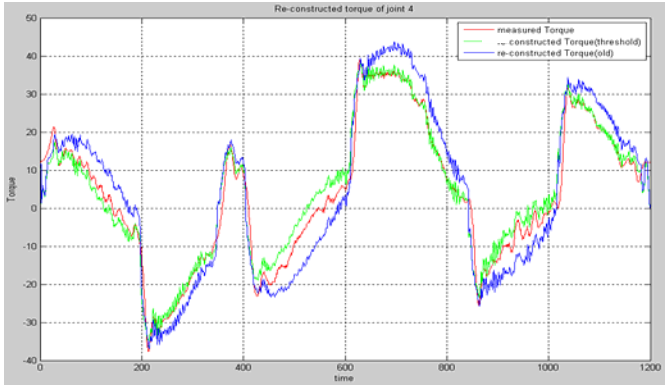
a) Joint 1



b) Joint 2



c) Joint 3



d) Joint 4

Figure 3
Measured torque vs. Re-constructed torque

The root-mean-square (RMS) errors between the measured torque and re-constructed torque are shown in Table 3.

Table 3
RMS errors between the measured and re-constructed torque

Joint	RT(thres): Set A	RT(all): Set B
1	4.1	**3.9**
2	**5.9**	7.1
3	**1.5**	2.1
4	**3.7**	5.8

Theoretically, one should expect the quality of the identified parameter $\hat{h}_b^A$ using set A is worse than the one using set B $\hat{h}_b^B$ because there are more data in set B. However, as can be seen in Table 2, an opposite result was obtained. The RMS errors in the first case (set A) are smaller than the second case (set B) for most of the joints. This observation implies that the extra data points in set B contribute negatively to the accuracy of the identified result $\hat{h}_b^B$ in low velocity regions [9].

3.2.2 Computed Torque Control

Since the purpose of the identification process in this paper is to obtain a model that can be used in advanced model based control schemes, the identified model has been further tested in another experiment as descrbed below:

1 All the joint(s) of the manipulator is commanded to follow a sinusoidal trajectory (amplitude: 30 degree, period = 4s).
2 Two controller schemes were implemented:
   a) *Independent joint control*: no dynamic information was used to compensate for the inertial effects. This control scheme is widely adopted in most industrial manipulator controller because of its simplicity.
   b) *Dynamic control*: the identified dynamic parameters were used. A standard joint space computed control was implemented. The identified dynamic model was used to decouple the dynamic behavior among the axes. Notice that the feed-forward frictions i.e. the compensated frictions are computed based on the desired joint velocities.

The tracking errors for the first four joints are shown in Figure 4: blue represents the joint tracking errors using the indepedent joint control scheme, red represents the joint tracking errors using the dynamic control.

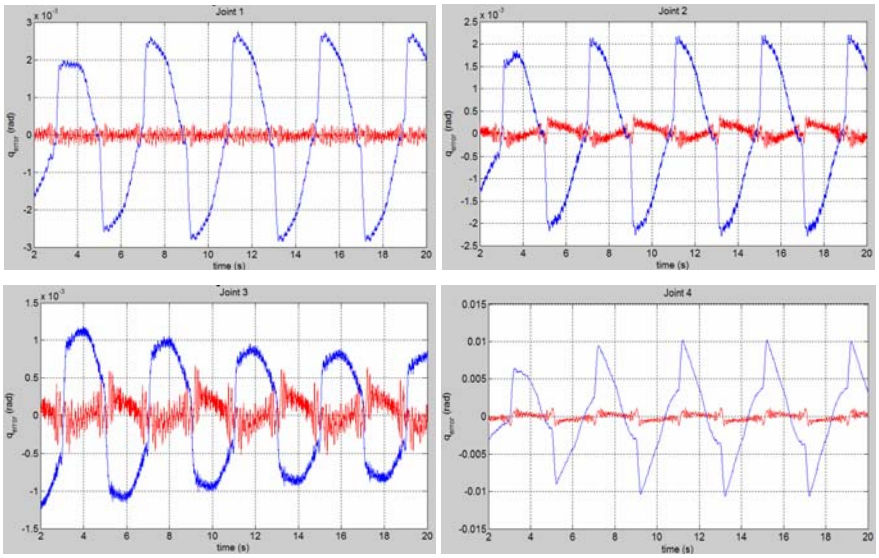


Figure 4
Joint tracking error comparison between kinematic and dynamic control (left to right, top to bottom: joint 1 to 4)

It is clear that there is a significant improvement in term of the tracking error for joint 1, 2 and 4. The tracking error for joint 3 is not as much different as others. One explaination that has been pointed out in [9] is because of the structure of the PA10 that makes the inertial effects at joint 1, 2, 4 are much easier to be excited than the rest of the joints. As a result, the quality of the identified parameters which contribute to the joint torque of joint 3 are poorer. This observation implies that further constraints need to be imposed on the optimum trajectories in order to excite the dynamic effects from different joints evenly.

It is worth pointing that the above identified dynamic model was obtained in the high speed region. Consequently, it is neccessary to see how good the identified model in the low speed region is. In order to check the performance of the identified parameters in the low speed region, the above experiment has been redone with the period of the desired trajectory incresed from 4 s to 40 s. Tracking errors are shown in Figure 5 (blue: indepedent joint control scheme, red: dynamic control).

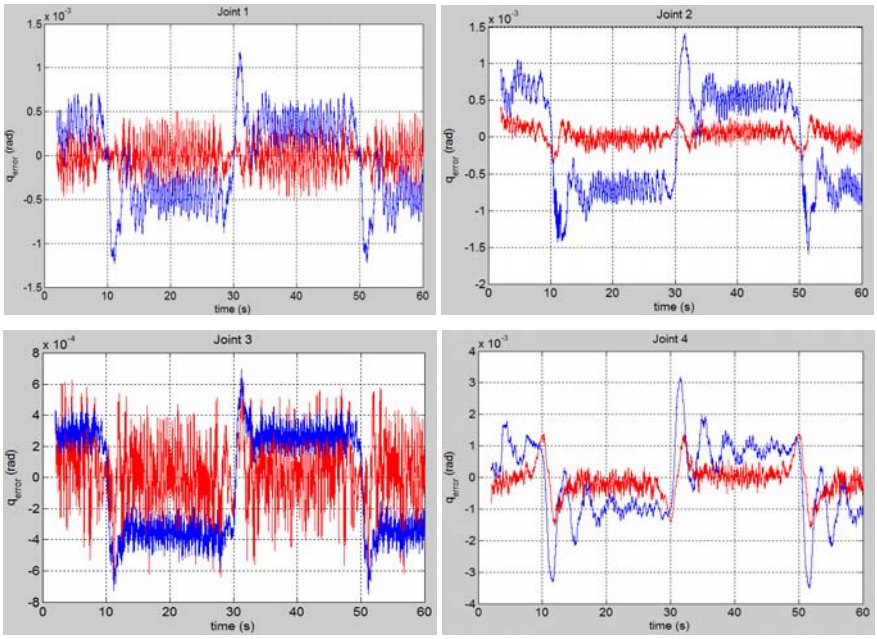


Figure 5
Joint tracking error (low speed) comparison between kinematic and dynamic control (left to right, top to bottom: joint 1 to 4)

As is seen, the differences between the model based control and non-model based control are no longer significant as in the previous case. One possible explaination is that the inertial effects of the dynamic model has been dominated by joint frictions at low speed region. As a result, the control performance will mainly depend on how joint frictions are compensated in this region.

Conclusions

We have presented a systematic procedure for identification the dynamic model of robot manipulator(s). Two main considerations has been addressed through the process. Firstly, the validity of the most commonly used joint friction model, the combination of viscous and Coulomb friction, was considered. Secondly, the problem of the so-called physical feasibility of the identified parameters has been mentioned. Instead of using the standard linear least-square estimator, a constrained optimization problem was used to obtain the identified parameters. The proposed approach has been implemented on the first four links of the Mitsubishi PA10 manipulator. The correctness of the identified dynamic model was verified by comparing the reconstructed torques from the identified model and the measured torques from each joint. Furthermore, the usefulness of the identified parameters has also been justified by incorporating the identified dynamic model to the conventional computed-torque control scheme. Significant

improvement in terms of the joint tracking errors was obtained in comparison to the one using non-model control scheme in the high speed region. In the low speed region, however, this observation is no longer true. Thus, a different model should be used to describe the dynamic behavior of the manipulator in this region.

Acknowledgements

The authors would like to thank the members of the Mechatronics group at SIMTech (Singapore Institute of Manufacturing Technology – www.simtech.a-star.edu.sg) who helped conduct the experiment on the PA10 manipulator.

References

[1] An, C. H., C. G. Atkeson, J. M. Hollerbach, *Model-based Control of a Robot Manipulator*. The MIT Press series in artificial intelligence. 1988, Cambridge, Mass.: MIT Press. p. 233

[2] Khatib, O., A Unified Approach for Motion and Force Control of Robot Manipulators: the Operational Space Formulation. IEEE Journal of Robotics and Automation, 1987. RA-3(1): pp. 43-53

[3] Armstrong, B., et al. *The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm*. 1986, San Francisco, CA, USA: IEEE Comput. Soc. Press

[4] Daemi, M., B. Heimann, *Separation of Friction and Rigid Body Identification for an Industrial Robot.* Courses and Lectures-International Centre For Mechanical Sciences, 1998: pp. 35-42

[5] Armstrong, B., On Finding Exciting Trajectories for Identification Experiments Involving Systems with Nonlinear Dynamics. International Journal of Robotics Research, 1989. 8(6): pp. 28-48

[6] Antonelli, G., F. Caccavale, P. Chiacchio, A Systematic Procedure for the Identification of Dynamic Parameters of Robot Manipulators. Robotica, 1999, 17: pp. 427-35

[7] Grotjahn, M., M. Daemi, B. Heimann. *Friction and Rigid Body Identification of Robot Dynamics*. 2001, Rio de Janeiro, Brazil: Elsevier

[8] Kermani, M. R., R. V. Patel, M. Moallem, *Friction Identification and Compensation in Robotic Manipulators.* IEEE Transactions on Instrumentation and Measurement, 2007. 56(6): pp. 2346-2353

[9] Ngoc Dung, V., J. Marcelo H. Ang Jr, Improved Dynamic Identification of Robotic Manipulators in the Linear Region of Dynamic Friction, in the 9th International IFAC Symposium on Robot Control 2009

[10] Gautier, M., W. Khalil, *Direct Calculation of Minimum Set of Inertial Parameters of Serial Robots.* IEEE Transactions on Robotics and Automation, 1990, 6(3): pp. 368-373

[11] Gautier, M., *Numerical Calculation of the Base Inertial Parameters of Robots.* Journal of Robotic Systems, 1991, 8(4): pp. 485-506

[12] Presse, C., M. Gautier. *New Criteria of Exciting Trajectories for Robot Identification*. 1993, Atlanta, GA, USA: IEEE Comput. Soc. Press

[13] Yoshida, K., W. Khalil, Verification of the Positive Definiteness of the Inertial Matrix of Manipulators Using Base Inertial Parameters. International Journal of Robotics Research, 2000, 19(5): pp. 498-510

[14] Ngoc Dung, V., J. Marcelo H. Ang, Dynamic Model Identification For Industrial Manipulator Subject To Advanced Model-based Control, in 4th International Conference Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management. 2008: Manila, Philippines

[15] Olsen, M. M., J. Swevers, W. Verdonck, *Maximum Likelihood Identification of a Dynamic Robot Model: Implementation Issues.* International Journal of Robotics Research, 2002, 21(2): pp. 89-96

[16] Mata, V., et al., Dynamic Parameter Identification in Industrial Robots Considering Physical Feasibility. Advanced Robotics, 2005, 19(1): pp. 101-19

[17] Khosla, P. K., *Categorization of Parameters in the Dynamic Robot Model.* IEEE Transactions on Robotics and Automation, 1989. 5(3): pp. 261-8

[18] Siciliano, B., O. Khatib, *Springer Handbook of Robotics*. 2007: Springer-Verlag New York, Inc.

[19] Gautier, M., N. CNRS. Dynamic Identification of Robots with Power Model. 1997

[20] Swevers, J., W. Verdonck, J. De Schutter, *Dynamic Model Identification for Industrial Robots.* Control Systems Magazine, IEEE, 2007. 27(5): pp. 58-71

[21] Swevers, J., et al., *Optimal Robot Excitation and Identification.* Robotics and Automation, IEEE Transactions on, 1997, 13(5): pp. 730-740

Towards a Globalized Software Industry

Hannu Jaakkola

Tampere University of Technology, Pori, Finland
hannu.jaakkola@tut.fi

Abstract: Software Engineering (SE) represents a remarkable share of the Information and Communication Technology (ICT) industry, which is an important feature of modern well-developed societies. Its importance can be seen on two levels: the industry itself and its indirect consequences for other industries. Although the SE industry represents a reasonably small share of employment (in the home country of the author, 2-3%), the indirect effects raise its economic importance to a much higher level. The role of the ICT industry is also used as one of the Information Society (IS) metrics in country comparisons. One of the leading trends of modern SE is globalization: employees represent different national cultures and the organizations are distributed across several countries. This paper opens the discussion on the topics worth considering when making globalization decisions. The basic hypothesis is that decisions are mainly based on economic factors. The organizational and cultural factors connected to these decisions are not well understood. At the beginning of the paper some background is introduced and the results of related studies are listed. The paper introduces a framework that is useful in planning the globalization of an organization. This paper focuses on the Finnish software industry; however, the same principles are applicable independent of the country or nation.

Keywords: cross-cultural, multicultural, software engineering (SE), cultural sensitivity, global software development, information and communication technology (ICT), globalization, global software development (GSD)

1 Introduction

1.1 The Economic Value of the Software Industry

The software industry is an important business sector – both on a worldwide scale and in the Finnish dimension. The total value of the global software market has been projected at EUR 238 Billion [1, 2]. The figures for Finland are as follows:

- In 2008 software product business revenues (composed of the software product business and related services) accounted for EUR 2.32 billion; about one-third of it from international business.

- The latest statistics show an 8.7% annual growth in business volume; the software industry tends to be reasonably independent of economic trends.
- The number of Information and Communication Technology (ICT) companies in Finland is 8 800 [3]; they employ 51 000 people, 33 000 of which are in the software industry.
- According to a recent report [4], two-thirds of software companies employ less than 20 employees; 72% of companies have an annual revenue of below EUR 20 million. The average revenue per employee is EUR 100,000.

However, the figures above do not tell the whole story, because the analysis is of the ICT industry in its narrowest sense. It does not cover the companies developing products controlled by embedded computers and software. More than pure software products, these represent the leading edge of the Finnish ICT industry. This product category includes instrumentation, telecommunications, and machine engineering products, among others. Typical examples of this category are mobile phones and harvesters. Nokia has reported that the development costs of mobile phone software represent approximately 80% of the total. A harvester used to harvest timber in forests is a high-tech product controlled by close to ten computers interacting with each other. The total size of the software in these products is in the range of millions of lines of code, representing a reasonably wide variety of software types, from very typical administrative applications to complex real time and telecommunications software.

Software engineering (SE) is demanding expert work. Typical of this kind of work is its indirect effect on employment. According to statistics, a software engineer typically employs 2-3 people in related jobs. In summary, the total employment effect of the software industry can be calculated more likely as 10% of employment rather than of a few percentage points only.

The importance of the software industry is also important on an indirect level. It is one of the information society metrics in country comparisons. The home country of the author is ranked high in international comparisons, which means that the information technology-based infrastructure is well developed and the industrial structure is highly dependent on the welfare of the ICT industry. For the sectors utilizing ICT-related products, the proximity of the ICT industry itself is crucial.

1.2 The Software Industry is Globalizing

One of the trends that is changing the characteristics of the software industry very strongly is *globalization*. The driving forces are manyfold and cover the factors derived from business goals, characteristics of the products, and changes in the software engineering development process. The factors connected to the *software business* are motivated by growth, access to wider markets, and the availability/price of the resources required. The current *software* products are

more products than individual artifacts. They are modular, adaptive, and based on *industrial* development methods. As a consequence, the distribution of work has become a natural part of the *development process*. Because of the "industrial overhead," software products are also growing in size and complexity as components of the *complex systems of systems;* as a result, their development and maintenance are based on distributed responsibilities. Even the traditions in the development culture are breaking down; the productivity of the traditional plan-oriented development has been impugned and an alternative approach is provided by the Agile approach (see e.g. [5]. In addition to traditional software (based on ownership of the product or licenses), different kinds of service-based solutions are overwhelming the market (e.g. SOA-based solutions, ASP- and SaaS-based services).

Globalization has meaningful consequences for the software industry:

- software companies are establishing branch offices abroad;
- software companies are offshoring their processes, outsourcing their work, and subcontracting abroad;
- ownership of companies is becoming global; therefore several Finnish software companies are today under foreign ownership and Finnish companies own companies abroad.

These factors require the easy flow of workforce over geographical borders and also meaningful changes in software organizations, which are becoming *multicultural*. Globalization may be a success or a failure.

This paper is based on an ongoing research project. The project has just started and its goal is to identify the factors that have an effect on successful globalization, as well as to find the factors behind failures. The focus is set on the development process, its consequences in the organization, management, and division of work, taking cultural aspects into account. However, product viewpoints (usability, localization, user aspects) are excluded. In sub-chapter 1.3 we start with two "real life" cases.

1.3 Two Scenarios

Scenario 1: A Finnish company (A) is establishing a branch office in China. The decision was based on the calculations of the salaries and availability of a skilled workforce. The purpose is to offshore a part of the software development to China (i.e. moving work from Finland to a cheaper country). The management of the company were not familiar with Chinese culture and lifestyle, but they expect to manage this problem because of their previous experience in international business. One of the experienced members of the higher management was relocated to China to establish the branch office – his responsibility was to build

up the infrastructure and recruit the first employees. Plans to distribute the project responsibilities were available and the organization has experience in intra-organizational distributed work culture. However, the opening of the office in Beijing was delayed by half a year due to the problems with *bureaucracy* and *local policy*. The banking system does not support free money transmissions. Additionally the first Chinese employees were not totally committed to the company and left after three months because of a better offer from an American company. After three years the organization was operative, but its costs exceeded the costs of the Finnish part of the company – the salary level was 80% of the salary paid in Finland. Additionally regular quality problems exist in the products developed in China – the processes applied in Finland do not fit Chinese working culture. Company (A) is considering closing the branch office in China because the availability of the workforce in Finland has also become easier and China is not a promising market area for the products of company (A).

Scenario 2: A Finnish company (B) was profitable as an important subcontractor of software and its main client was a big Finnish globally-operated company (D). Part of the work was done in collaboration with an Indian subcontractor (C), which also collaborated with D. Thus, the organization has been used to collaborating in cross-cultural teams. The Indian company (C) is a large one (30,000 employees) and operates in diversified fields of business, software development being only one of them. C offered to buy the Finnish company (B); after six months of consultation, the agreement was finalized and the Indian company C became the owner of B; a part of the agreement covered the willingness to continue and extend operations in Finland. After purchase, some Indian experts were moved to the Finnish offices of B. Finnish management is continuing but the budget and growth expectations are set by Indian head office – at a far too demanding level in Finnish circumstances, which are not familiar to the Indian management. After two years, a part of the operations that had been under the responsibility of the Finnish part of the joint organization (B+C) was moved to India and half of the employees were faced with two alternatives - either to move to India or leave the company.

1.4 Study Problem

The scenarios above provide a rather pessimistic view of globalization. Apart from some negative experiences, most cases are, however, more or less successful – some great successes and some leading to an acceptable final state. This paper is based on the background ideas of a reseach project (called STEP) recently started by the author and his research group. Its *primary goal* is to deepen the understanding of the globalization process of software organizations. The cultural aspects are mental and not easy to anticipate. The study covers two viewpoints: (1) a *Finnish company* that is extending its activities to foreign cultures or (2) is as a

result of acquisition, now under the ownership of a company representing a foreign culture. The study is based on the following hypothesis:

1 Globalization decisions are made on economic grounds and prior knowledge of the foreign culture is usually poor when the globalization decision is made.

2 The elements of successful globalization vary according to the organizational characteristics, direction of globalization, the role of the owner in the global collaborative network, and the globalized artifact.

3 The culture-based stereotypes recognized by Hofstede (the selected framework of this study) are helpful in managing cultural aspects but need additional interpretation in order to be used in the context of the software industry.

The research is focused on *organizational and process issues* (how the global collaboration is organized and how aspects derived from the multicultural characteristics of the organization are taken into consideration). The viewpoints of product localization and user experience in different cultures are excluded.

The work is just in its initial stages and it is too early to provide any results in this paper. The study itself contains two phases of interviews: the first phase (target group is selected experts in the field) is to set the focus of the second phase (target group is a statistically meaningful set of software companies) interview to the right scope. In addition, a relevant literature survey has been executed to become acquainted with the existing findings. Surprisingly, there are not very many reported results available on the topic. The interview results will be analyzed using the framework introduced in this paper. At the moment the first interview phase and literature survey is ongoing. This paper concentrates on the current findings and introduces the methodology applied in the analysis.

1.5 Structure of the Paper

The purpose of this paper is to analyze the factors related to software engineering in a cross-cultural and also in most cases in a distributed environment. Section 2 introduces the frameworks developed to recognize cultural differences. Section 3 lists the findings of related studies, mainly based on real cases analyzed and reported in SE journals and conferences. Section 4 introduces the analysis model that is used to simplify the complexity of the problems related to cross-cultural software engineering in globally operated organizations. The last section concludes the paper and lists activities for future work.

2 Analyzing the Cultural Differences

Terminology within the multicultural and cross-cultural communication field seems to be inconsistent. The concepts relevant to this study are defined below. The terminology is a part of a wider glossary of relevant terminology [6]:

> *Culture* is a collective phenomenon. It is shared with people who live or have lived within the same social environment, which is where it was learned. Culture consists of the unwritten rules of the social game. It is the collective programming of the mind that separates the member of one group or category of people from others [7, 8].
>
> *Cross-cultural* describes comparative knowledge and studies of a limited number of cultures [9].
>
> *Multicultural* describes comparative knowledge and studies of relating to, or including several cultures [9].

According to [7, 8] cultures can be considered on several levels: national, organizational, sub-organizational, professional, domain, project, team and task cultures (see e.g. [10], [11]). In this paper, the focus is on *multicultural* organizations and on the analysis of the role of *national cultures* in multicultural software organizations.

There are two frameworks of the analysis of the differences of national cultures that are widely used and applied. The most referred to is the work of Hofstede [7, 8, 13]. The framework (summary in Table 1) recognizes five properties that separate cultures from each other (see the figure enclosed [12]). *Individualism/ Collectivism* (IND) describes the extent to which a society emphasizes the individual or the group. *Power Distance* (PDI) describes the extent to which a society accepts that power is distributed unequally. *Masculinity/Femininity* (MAS) refers to the values to be held in a society. *Uncertainty avoidance* (UAI) refers to the extent that individuals in a culture are comfortable (or uncomfortable) with unstructured situations. *Long-term/Short-term orientation* (LTO) refers to the extent to which a culture programs its members to accept delayed gratification of their needs.

Hofstede's www-page [7] includes a useful tool to compare cultural differences. It provides an opportunity to study the characteristics of individual cultures or to analyze differences between two (or more) cultures (Figure 1).

Table 1
Hofstede's Cultural Dimensions [12]

Dimension	Description of the dimension
Individua-lism/ Collectivism IDV	Individualism/Collectivism describes the extent to which a society emphasizes the individual or the group. Individualistic societies encourage their members to be independent and look out for themselves. Collectivistic societies emphasize the group's responsibility for each individual.
Power distance PDI	Power distance describes the extent to which a society accepts that power is distributed unequally. When the power distance is high, individuals prefer little consultation between superiors and subordinates. When the power distance is low, individuals prefer consultative styles of leadership.
Masculinity/ Femininity MAS	Masculinity/Femininity refers to the values more likely to be held in a society. Masculine societies are characterized by an emphasis on money and things. Feminine cultures are characterized by concerns for relationships, nurturing, and quality of life.
Uncertainty avoidance UAI	Uncertainty avoidance refers to the extent that individuals in a culture are comfortable (or uncomfortable) with unstructured situations. Societies with high uncertainty avoidance prefer stability, structure, and precise managerial direction. In low uncertainty avoidance societies, people are comfortable with ambiguity, unstructured situations, and broad managerial guidance.
Long-term/ Short-term orientation LTO	Long-term/Short-term orientation refers to the extent to which a culture programs its members to accept delayed gratification of their material, social, and emotional needs. Business people in long-term oriented cultures are accustomed to working toward building strong positions in their markets and do not expect immediate results. In short-term oriented cultures the "bottom line" (the results of the past month, quarter, or year) is a major concern. Control systems are focused on it and managers are constantly judged by it.

In Finland the most promising reference cultures are China and India as the most feasible goals for offshoring/outsourcing. Figure 1 shows the differences between the cultures – the facts that must be taken into account in organizing the work and in management of a multicultural organization. Finns are used to working in low democratic organizations (low PDI value) as individuals (high IDV value), whereas Chinese and Indians respect the power and status based on the hierarchy – this has direct consequences e.g. in decision making and in liability; they are also used to acting as a member of social groups both in work and private life. Finnish culture is family-oriented and feministic, whereas Chinese and Indian values are categorized as more masculine (MAS; money, property). This has reasonable high consequence on the factors motivating people to work. Finns accept higher uncertainty (UAI) than the reference cultures, in which "losing face" is not acceptable. Indian and Chinese cultures are typical examples of long-term orientation – the ability to wait for the results of the work is high. This metrics value is not available for Finland, but an example of short-term orientation is the USA.

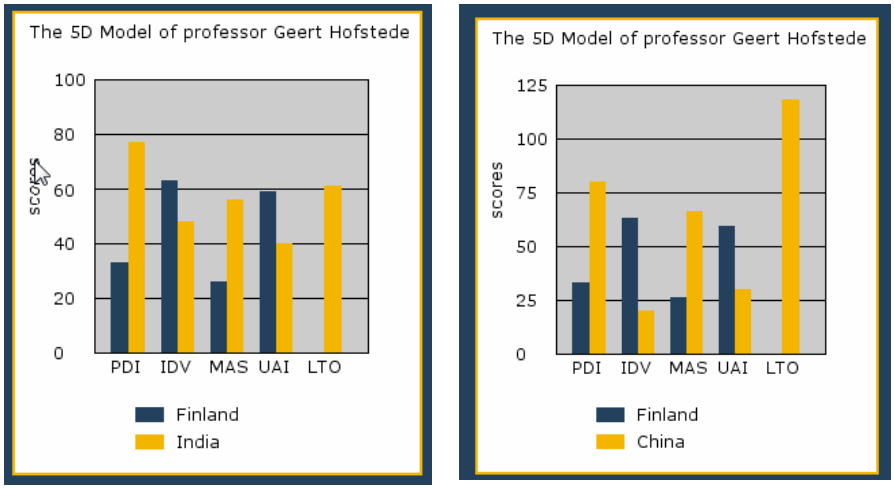


Figure 1
Comparison of Finland to China and India according to Hofstede's model

Another widely used model to analyze national cultures is the Lewis Model [9]. It focuses more on communication and interaction skills, and cultures can be classified into three main categories: a *Linear-active culture* is task-oriented and value is given to technical competence; a *Multi-active culture* is extrovert and human force is seen as an inspirational factor; a *Reactive culture* is people-oriented and dominated by knowledge, patience and quiet control. National cultures are located on the sides of the triangle having the three above-mentioned stereotypes as the corners [14]. The people representing *linear-active* cultures are communicative, but not too much. They concentrate on one activity at a time and are used to working step-by-step. They are polite but direct, job-oriented and are grounded on facts. *Multi-active* cultures talk all the time, used to the concept that parallel activities and plans provide more outlines than rules. They respect feelings more than facts and their "truth" is flexible. People from *reactive* cultures are listeners and activated as a reaction to the partner's action. They are indirect and diplomatic, and very people-oriented.

The frameworks have been widely criticized – especially by researchers in social sciences. It is clear that the analysis based on the wide material collected by means of interviews is not the final truth. It is also true that every culture is a set of individuals and not a homogenous one. However, the stereotypes identified by the models help to recognize some basic rules on how to organize the work of a culturally heterogenous group of people. Organizing the work, dividing the activities, adapting the management to take into account the cultural differences are examples of multicultural leadership and management culture.

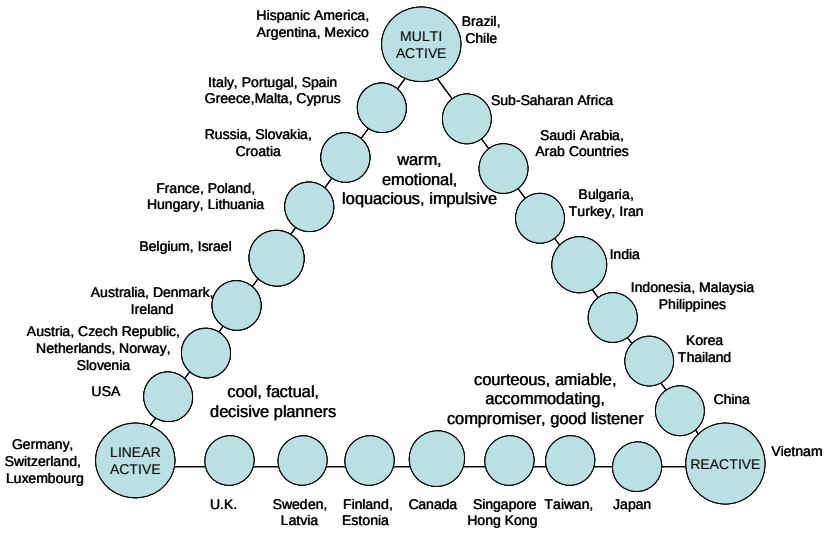


Figure 2
Cultural Types by Lewis

3 Multicultural Software Engineering – Current Findings

In spite of the fast-growing tendency towards globally distributed collaborative multicultural organizations in SE, there are just a few objective reports publicly available. Relevant sources are available from SE journals (ACM, IEEE, Springer, etc.) and conferences (e.g. ICSE – International Conference in Software Engineering and ICGSE – International Conference on Global Software Engineering, among others). The most commonly applied reference model is Hofstede's model. The findings (see Table 2) cover the differences found and provide recommendations in the area of practices related to software life cycle management, contracting, attitude to working time, mental mode (attitude to bureaucracy, authorities, the role of values and norms), meeting practices, team work, feedback practices, expectations in communication, division of work, importance of specifications, risk management, product management. Additionally, the papers cover recommendations of how to take multicultural aspects into account in SE education.

Table 2
Summary of the results of the selected studies in multicultural SE

Source	Reference model	Main findings
[15]	Hofstede	*Cultures*: Indians and Non-Indians *Findings*: Different practices and attitude in software life cycle management and contents of the life cycles. The main findings focus in contracting, organizing the life cycle phases, attitude to working time, meeting practices, team work, feedback practices, expectations in communication and risk management.
[16]	Bloom's taxonomy [17]	*Cultures*: General *Findings*: Analysis of the the requirements set by multicultural context for university.
[18]	Hofstede	*Cultures*: Japan, India; America *Findings*: PDI, UAI, IDV indices in focus. Analyzing the meaning of the differences recognized in attitude to work, software architecture, division of work, product management
[19]	No	*Cultures*: General *Findings*: More focused generally on distributed software development. Requirements for SE education are from distributed and multicultural character of the organizations. Architecture and modular structure in focus.
[11]	Nonaka & Takeuchi Spiral	*Cultures*: General study of the role of cultures *Findings*: Cultural aspects in SE – views in development and education. Cultural aspects are seen as a context (among others). Analysis of different levels of cultures and application of Knowledge Creation Spiral by [20] to analyze the structure of SE and SE education.
[21]	Nonaka & Takeuchi Spiral; Boehm Spiral	*Cultures*: General study of the role of cultures *Findings*: Three-layer model (development process, knowledge & context, multicultural services) to manage the complexity of multi cultural SE (seen as knowledge work).
[12]	Hofstede Lewis	*Cultures*: General study of the role of cultures *Findings*: An overview of applicable frameworks in cross-cultural software development environment and synthesis of the existing research findings. Three factors: distribution, cultural differences and ownership of the network.
[22]	Hofstede	*Cultures*: US, India, Western Europe, Japan *Findings*: Analysis of real outsourcing cases. Difference in agreement culture, level of expected documentation and in the mental mode of the cultures (attitude to bureaucracy, authorities, the role of values and norms etc.). The problems arising inside cross-cultural teams are different than those arising inside teams representing the same nationalities / cultures / language groups. The beneficial use of "bridging teams" was seen as important, as well, to unify the organizational culture in the long term.
[23]	Hofstede	*Cultures*: General *Findings*: Culture has an important role in the successful adoption

		of Total Quality Management (TQM) in an organization. The result is a framework that can be used for further analysis.
[24]	No	*Cultures*: Several – not specified. *Findings*: Results of a case study, in which the cultural mix of SE design teams was analyzed. The members of the teams were undergraduate students of a university-level IT curriculum. The main finding is that in teams, the cultural strengths of some members support the weaknesses of the others.
[25]	Structuratio nal analyses, Hofstede;	*Cultures*: Jamaica, India (Case studies) *Findings*: A theoretical framework for cross-cultural aspects in SE. The paper reports two case studies, one from Jamaica and one from India. The framework is based on a structurational analysis method, which is compared to Hofstede's model and the findings of two offshoring cases.
[26]	Pattern approach	*Cultures:* General *Findings:* Recognized project patterns to be applied in global software development projects. Applicable patterns and a pattern language are introduced.

The studies published are introductory rather than the analysis of existing cases. There has been no effort to generalize the results and by comparisons to find some common phenomena to act as a guideline for planning globalization activities. In our studies, as a part of an ongoing research project, we have recognized the complexity of the problem and tried to find ways to structuralize it. This framework is explained in Section 4.

4 Five Factor Model for Analyzing Multicultural SE

The Five Factor Model (FFM) is based on the factors of globalization introduced in [12]. These factors are considered to be important in analyzing the data collected through interviews. The five-dimensional space provides a means to classify the data and find differences/similarities between the categories recognized.

The relevant factors of the FFM are:

1 Organizational characteristics of distribution and globalization (Table 3);

2 Frameworks of cultural differences – existing knowledge of cultural differences (Section 2 of this paper);

3 Direction of the globalization (classification factor);

4 Ownership of the global organization/process (classification factor);

5 Artifact: product, process, service (classification factor).

Table 3
Organizational dimension

A.Outside	Broker	Broker Network	Broker Offshoring	Broker Outsourcing	Traditional Subcon-tracting
B. Inter-Organi-zational	Virtual Organiza-tion	Distributed Virtual Organiza-tion	Traditional Offshore	Traditional Outsourcing	NULL
C. Intra-Organiza-tional	Traditional	Distributed Traditional	Concern based Offshore	NULL	NULL
	1. One site	**2. Multiple sites**	**3. Offshore**	**4. Outsource**	**5. Subcon-tract**

The model will be used to classify the findings that will be collected from the globalization experiences of Finnish software companies.

Organizational characteristics of the global organization are classified according to Table 3 (Globalization Grid). The left-to-right dimension lists the type of distribution and the down-to-up dimension the characteristics of the organization. In the figure, the companies inside the "marked" area are the focus of the project related to this paper; the 42 companies representing different categories have been selected and will be interviewed as part of the project. The article of Herbsleb and Moitra [27] handles briefly the role of distribution in global software development. They point out that SE requires a lot of communication – both formal and informal. The former needs a clear well-understood interface. However, the role of informal communication is emphasized especially when uncertainty increases in the projects. Outsourcing and offshoring typically decrease the opportunity for this kind of communication, if it is not especially taken into account by providing tools to support it.

In our work the Hofstede Model is used as the *framework for cultural analysis* (because of its frequency in related studies, despite knowing the *critique against it*). The aim is to test the validity of the recognized stereotypes in the SE industry.

The globalization direction is an important factor. It tests the similarities/ differences in the case where a Finnish company (or employee) is going into a foreign culture and vice versa. The expectation is that the same rules will not apply in both cases.

The ownership factor tests the ruling mechanism in the global network – this has consequences in adaption mechanisms either for the national culture or the owner's culture – two approaches, permissive (adaptation into one culture) and non-permissive (unification into one culture) in adaptation is studied.

The *artifact factor* studies the differences in the globalization of product development, processes (process over cultural borders) and services (over cultural borders). A sixth dimension – not included here – could be added to study the differences in the *organizational status* of the employees (expert, project management, higher management).

Conclusions and the Future Work

Based on the first phase interviews, some common findings are available. In the reflection of an organization and national cultures we have recognized two approaches – *adaptive* and *permissive*. In practice, we either have to adapt processes or people. Especially in big firms, the employees are recruited as individuals, not as members of certain national cultures. The adaptive approach is used and new staff members have to adapt to the *company culture*. Their career is also culture-independent. The main benefit of this approach is that the same operative processes are applicable all over the company and the employees are easily transferrable from one organizational part to another. In the case of competition of a skilled workforce, this approach will increase the loyalty of employees to the company and bind them more tightly to the organization than in the case of the permissive approach (i.e. employees' cultural background is taken into account and the processes are adapted to it). Experience also shows that employees who do not like the "company way" will leave the company quite soon. The disadvantage of the adaptive approach is that the strengths of the cultural background are not benefited from.

The cases analyzed to date all relate to a situation where a Finnish organization has established activities in lower labor-cost countries (China). The differences in the economic situation between Finland (high standard of living; *mature culture*) and the target country (lower standard of living, lower salary level; *emerging culture*) causes differences in work motivation: the marginal benefit of extra work in the emerging culture (Hofstede's masculine values) is higher than in the mature culture (where feministic values are ranked higher). Independently, the motivation factors seem to remain the same because the surrounding society does not change as much as the company culture.

Experiences show that the direct cost difference is only a temporary benefit. In a few years, the salaries in lower labor-cost countries tends to approach the level valid in the host country. It may also mean that people are ready to move to the target country without extra benefits (e.g. extra financial support for moving from Finland to China to work in the "China office").

There are also scenarios that discuss the division of the work between higher and lower salary countries. One of the potential futures is that the importance of the first phases (front-end engineering processes) of the software development life cycle will grow. It means that the division of work in a globally distributed organization leaves front-end processes (requirements elicitation and specification, design) in host (higher salary) countries and the back-end processes are the responsibility of the outsourced parts of the organization located in lower salary countries. Partly, this is already the situation today. There are also opposite examples that are based on good experiences in product conceptualization and product design e.g. in China.

This field of study has not yet been widely researched and the results available are mainly non-generalized experience reports. Multicultural working is demanding and, to manage the related problems, it covers topics in organizational studies, management (person, organization, project and team), and the social environment. As a study topic on the one hand it is challenging, but on the other hand it provides a new understanding of complicated social collaborative networks. Globalization studies are also culture-dependent – studies concerning Finnish organizations are not yet available. In the project related to this paper the aim is to produce new knowledge directly applicable to the Finnish software industry. Our expectation is that the results will also be applicable to other cultures, if the cultural differences are recognized and understood. This project will also open new research questions for further studies and provide a forum for permanent cross-cultural researcher networks. The findings will also be integrated into the SE curriculum – first in the applicant's own organization, but also probably on a wider scale.

Acknowledgements

This work is supported by the Finnish Funding Agency for Technology and Innovation (Tekes) and the Academy of Finland.

References

[1] EITO, *European Information Technology Observatory - EITO 2007 Report*, EITO, 2007

[2] M. Rönkkö, J. Ylitalo, J. Peltonen *et al.*, *National Software Industry Survey 2009*, Espoo, 2009

[3] M. Rönkkö, O.-P. Mutanen, N. Koivisto *et al.*, *National Software Industry Survey 2008: The Finnish Software Industry in 2007*, Espoo, 2008

[4] H. Kuitunen, J.-P. Jokinen, A. Lassila *et al.*, *Finnish Software Product Business: Results from the National Software Industry Survey 2005*, Centre of Expertise for Software Product Business and Tekes, Espoo, 2005

[5] Boehm, and R. Turner, *Balancing Agility and Discipline: A Guide for the Perplexed*: Addison-Wesley Longman Publishing Co., Inc., 2003

[6] A. Heimbürger, H. Jaakkola, T. Ylikotila, *Terminology of Multicultural Software Business. Internal Report of STEP Project.*, University of Jyväskylä and Tampere University of Technology, Jyväskylä and Pori., 2009

[7] G. Hofstede. "Cultural Dimensions," http://www.geert-hofstede.com

[8] G. Hofstede, G. J. Hofstede, *Cultures and Organizations: Software of the Mind: Intercultural Cooperation and Its Importance for Survival*, p.^pp. 300, New York: McGraw-Hill, 2004

[9] R. D. Lewis, *When Cultures Collide. Managing Successfully Across Cultures*, London: Nicholas Brealey Publishing, 1999

[10] W. R. King, "A Research Agenda for the Relationships between Culture and Knowledge Management.," *Knowledge and Process Management,* Vol. 14, No. 3, pp. 226-236, 2007

[11] H. Jaakkola, A. Heimbürger, J. Henno, "The Roles of Knowledge and Context in Context-Aware Software Engineering - in Terms of Education and Communication " in MIPRO 2009, Opatija, Croatia, 2009, pp. 224-230

[12] H. Jaakkola, A. Heimbürger, "Cross-Cultural Software Engineering - Interkulturalni softverski inzenjering (Abstract)," in International Conference on Society and Technology 2009, Zadar, Croatia, 2009, pp. 40-41

[13] G. Hofstede, *Culture's Consequences, Comparing Values, Behaviors, Institutions and Organizations across Cultures.*, Thousand Oaks, CA: Sage Publisher, 2001

[14] R. D. Lewis. "Richard Lewis - Cross-culture," November 20, 2009; http://www.cross-culture.com

[15] L. R. Abraham, "Cultural Differences in Software Engineering," in Proceedings of the 2nd annual conference on India Software Engineering Conference Pune, India, 2009, pp. 95-100

[16] M. W. Birrig, K. J. Filipski, "Teaching Multicultural Perspective in Software Engineering," *Proceedings of the 37th ASEE/IEEE Frontiers in Education Conference*, pp. S4E1-S4E6: IEEE, 2007

[17] B. Bloom, *Taxonomy of Educational Objectives. The Classification of Educational Goals: Handbook I, Cognitive Domain*, New York: Longmans, Green, 1956

[18] G. Borcheres, "The Software Engineering Impacts of Cultural Factors on Multi-cultural Software Development Teams.," *Proceedings of the 25th International Conference on Software Engineering (ICSE '03)*, pp. 540-545: IEEE, 2003

[19] M. Hawthorne, D. E. Perry, "Software Engineering Education in the Era of Outsourcing, Distributed Development, and Open Source Software: Challenges and Opportunities," *Proceedings of the 27th International Conference on Software Engineering (ICSE '05)*, pp. 643-644: ACM, 2005

[20] I. Nonaka, H. Takeuchi, *The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation*: Oxford University Press, 1995

[21] H. Jaakkola, A. Heimbürger, P. Linna, "Knowledge Oriented Software Engineering Process in Multi-Cultural Context," *Software Quality Journal,* Vol. Accepted, 2009

[22] S. S. Krishna S., Walsham G., "Managing Cross-Cultural Issues in Global Software Outsourcing," *Communications of the ACM,* Vol. 47, No. 4, pp. 62-66, 2004

[23] K. V. Siakas, E. Georgiadou, C. Sadler, "Software Quality Management from a Cross-Cultural Viewpoint," *Software Quality Journal,* Vol. 8, No. 2, pp. 85-95, 1999

[24] A. L. Simcock, "Does a Multicultural Mix Bring an Extra Dimension to Software Engineering Design Teams?," *Global Journal of Engineering Education,* Vol. 2, No. 3, pp. 263-270, 1998

[25] G. Walsham, "Cross-Cultural Software Production and Use: A Structurational Analysis," *MIS Quarterly,* Vol. 26, No. 4, pp. 359-380, 2002

[26] A. Välimäki, J. Kääriäinen, K. Koskimies, "Global Software Development Patterns for Project Management," *Software Process Improvement. 16th European Conference, EuroSPI 2009, Alcala (Madrid), Spain, September 2-4, 2009. Proceedings* R. V. O'Connor, N. Baddoo, J. C. Gallego *et al.*, eds., pp. 137-148, Berlin, Heidelberg: Springer 2009

[27] J. D. Herbsleb, D. Moitra, "Global Software Development," *IEEE Software,* Vol. 18, No. 2, pp. 16-20, 2001

On the Combinatorial Characterization of Fullerene Graphs

Tamás Réti

Budapest Tech
Bécsi út 96/B, H-1034 Budapest, Hungary
reti.tamas@bgk.bmf.hu

István László

Department of Theoretical Physics, Institute of Physics
Budapest University of Technology and Economics
H-1521 Budapest, Hungary
laszlo@eik.bme.hu

Abstract: In order to characterize and classify quantitatively the local topological structure of traditional fullerene graphs a new method has been developed. The concept is based on the introduction of a finite set of novel topological invarians called pentagon arm indices. The definition of pentagon arm indices is similar to that of well known pentagon adjacency indices, and their common features is that both of them characterize the local topological neighborhood of pentagons included in traditional fullerenes. It will be demonstrated that pentagon adjacency indices and pentagon arm indices together can be successfully applicable for preselecting the stable candidates of lower fullerene isomers C_n with $n \leq 70$.

Keywords: graph invariant, pentagon-neighbor signature, prediction of fullerene stability

1 Introduction

Fullerenes are defined as 3-valent (3-regular) polyhedral graphs having only pentagonal and hexagonal faces.

Methods for topological characterization of fullerene isomers have made a steady progress over the past decade and many calculations of stabilities of traditional and non-traditional fullerenes have been reported [1-14]. A promising trend in fullerene science is the employment of graph theoretical invariants for the characterization of combinatorial structure and prediction of their stabilities.

Several topological descriptors have been proposed to evaluate and classify the topological structure of fullerene isomers: the pentagon adjacency index N_P [1-4], the Wiener index W_I [5], the resistance distance R_T [5], the Kekulé structure count [6], the graph independence number [7], the number of spanning trees [8], the combinatorial curvature [9], the bipartivity measure of fullerene graphs [10], the occurrence number of different structural motifs in fullerenes [12-14].

In the majority of cases, for the stability prediction of lower fullerene isomers C_n with $n \leq 70$ the pentagon adjacency index N_P (the so-called minimal-N_P criterion) is used [1, 3, 4]. Determination of the pentagon adjacency index N_P is based on the pentagon-neighbor signature $\{p_0, p_1, p_2, p_3, p_4, p_5\}$, where each entry p_k (k=0,1,2,…5) counts those pentagons that have exactly k pentagonal edge-neighbors. From these data the pentagon adjacency index N_P can be simply computed:

$$N_P = \frac{1}{2}\sum_k k p_k \tag{1}$$

where $\sum p_k = 12$. It is obvious that N_P is also equal to the number of edges between adjacent pentagons, in other words N_P is identical to the total number of fused pentagon pairs in an isomer.

According to the minimal-N_P rule it is supposed that fullerenes which minimize N_P are more likely to be stable than those that do not [3-5]. Consequently, it is believed that the buckminsterfullerene is the most stable C_{60} fullerene, because this is the only one for which N_P has a minimum value (N_P=0).

However in some cases the discriminating power (i.e. the efficiency of prediction) of N_P index is limited. (The minimal-N_P criterion does not suffice to uniquely characterize the structure of fullerene graphs with identical pentagon adjacency indices.) Even some lower fullerene isomers C_n with $n \leq 70$ are characterized by the same pentagon adjacency index N_P. In such cases, using N_P, the accuracy of stability prediction is problematic. For example, among C_{66} fullerenes there are three isomers with the same lowest pentagon adjacency index (N_P=2), moreover, among C_{68} fullerenes there exist 11 isomers with N_P=2.

In order to improve the efficiency of stability prediction, a novel three-variable topological descriptor denoted by Ψ has been constructed. This includes the N_P index, and additionally two other independent topological graph invariants as well. The construction of this novel descriptor Ψ is based on the introduction of the so-called pentagon arm signature vector, whose components can be simply computed from Schlegel diagrams of fullerenes.

2 Pentagon Arm Indices as Graph Invariants

In a fullerene a pentagonal face F_P has 5 vertices, and each vertex is incident to an edge not belonging to the pentagon under consideration. An edge E incident to a vertex of F_P is called an arm of F_P if i) both end-vertices of edge E are incident to pentagons, and ii) E shares two neighbor hexagons. This definition implies that any pentagonal face may have q=0, 1,2,…5 arms. Let us denote by n_q the number of pentagons having q arms in a fullerene. It follows that each fullerene can be characterized by a pentagon arm signature vector $\{n_0, n_1, n_2, n_3, n_4, n_5\}$, where each entry n_q (k=0,1,2,…5) counts those pentagons that have exactly q arms. Starting with this concept, for an arbitrary fullerene we define a pentagon arm index N_A as follows:

$$N_A = \frac{1}{2}\sum_q q n_q \tag{2}$$

where $\sum n_q = 12$.

From this concept it follows that parameter N_A is identical to the total number of edges whose end-vertices are incident to pentagons, and share two neighbor hexagons, exactly. It can be verified that for topological invariant N_{PA} defined as

$$N_{PA} = N_P + N_A \tag{3}$$

the inequality $0 \le N_{PA} \le 30$ holds [15]. Concerning the upper bound, it follows that for fullerene C_{20} (represented by the dodecahedron) $N_{PA}=30+0=30$, and for the buckminsterfullerene $N_{PA}=0+30=30$ holds. It is conjectured that for any other fullerenes the inequality $0 \le N_{PA} \le 25$ is valid. (For fullerene isomers C_{30} : 1(D_{5h}) and C_{50} : 271 (D_{5h}) we have $N_{PA}=N_P + N_A= 25$).

In order to construct the topological descriptor Ψ and classify the fullerene isomers into disjoint subsets, we used the first and second moments (M_1 and M_2) of pentagon arm signatures $\{n_0, n_1, n_2, n_3, n_4, n_5\}$:

$$M_k = \frac{1}{12}\sum_{q=0}^{5} q^k n_q \tag{4}$$

where k=1 and k=2, respectively. From the previous consideration it follows that $M_1=N_A/6$. By means of moments M_1 and M_2, the variance of q can be calculated as $VAR=M_2 - M_1*M_1$. It is easy to see that VAR =0 if and only if there exists a positive integer $0 \le q \le 5$ among the components of pentagon arm signature vector for which n_q =12 holds. Starting with this concept, a fullerene is called balanced (more exactly q-balanced) if there exists a non-negative integer q for which n_q =12 valid. This means that a fullerene is balanced if and only if, VAR=0 holds. In Fig. 1 some examples are given for balanced fullerenes.

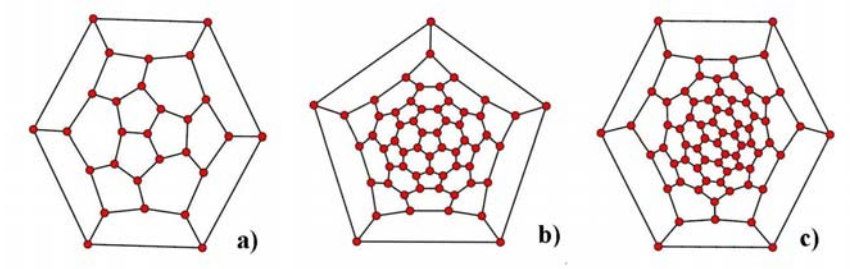


Figure 1
Schlegel diagrams of balanced fullerenes: a) 0-balanced C_{28} (T_d) isomer, b) 5-balanced C_{60} (I_h) isomer (buckminsterfullerene) and c) 4-balanced C_{72} (D_{6d}) isomer

The 5-balanced and 4-balanced isomers illustrated in Fig. 1 belong to the family of IPR (isolated-pentagon rule) fullerenes. It is known that the number of IPR fullerenes (fullerenes with N_P=0) is infinite. Fullerenes with N_{PA}= N_P + N_A=0 are called strongly isolated fullerenes. This definition implies that strongly isolated fullerenes represent a subset of IPR fullerenes. The number of strongly isolated fullerenes is also infinite. In Fig. 2 the Schlegel diagram of a strongly isolated fullerene is shown.

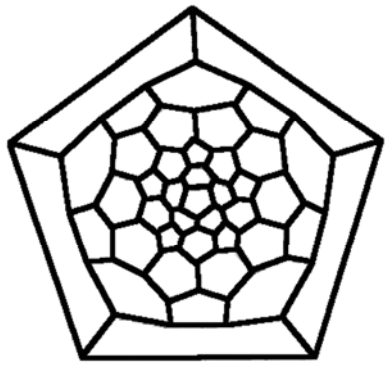


Figure 2
Schlegel diagram of the strongly isolated fullerene isomer C_{80} (I_h)

It can be verified that C_{80} (I_h) with vertex number 80 is the smallest strongly isolated fullerene. (See Fig. 2)

3 A Novel Graph-Theoretic Invariant for the Characterization of Fullerene Structures

In order to characterize the local combinatorial structure of fullerenes more efficiently, we defined the topological descriptor Ψ as follows

$$\Psi = \frac{30 + 6M_1}{1 + 4.5N_P + C(M_1, M_2)} = \frac{30 + N_A}{1 + 4.5N_P + C(M_1, M_2)} \tag{5}$$

where

$$C(M_1, M_2) = \frac{\sqrt{\frac{120M_2}{1+7M_1}}}{1 + 0.9(M_2 - M_1^2)^{1/5}} = \frac{\sqrt{\frac{120(VAR + M_1^2)}{1+7M_1}}}{1 + 0.9VAR^{1/5}} \tag{6}$$

For balanced fullerenes, (where VAR=0), coefficient $C(M_1,M_2)$ can be rewritten it the following simplified form:

$$C(M_1, M_2) = M_1\sqrt{\frac{120}{1+7M_1}} = N_A\sqrt{\frac{20}{6+7N_A}} \tag{7}$$

As can be seen, Ψ is defined as a function of 3 algebraically independent graph invariants: the pentagon neighbor index N_P and the moments M_1 and M_2. It follows that for strongly isolated fullerenes $N_P = N_A = C(M_1,M_2) = 0$, consequently in this case Ψ=30. The constants included in Eqs. (6 and 7) were estimated using numerical methods, as a result of analyzing the possible combinatorial structures and the energetic parameters of C_{40} isomers. This choice is explained by the fact that several topological descriptors have been already calculated for C_{40} fullerene isomers.

As it is known isomer C40:38 is predicted to be the C_{40} fullerene of lowest energy by many methods [3-5], this is followed by C40:39 and C40:31 isomers. It has been also shown that C40:38 fullerene has the lowest resistance-distance in the set (R_T=920,27). Two C_{40} isomers (C40:38 and C40:39) have the smallest pentagon adjacency indices (N_P=10). Among the 40 isomers of C_{40}, fullerene C40:1 is the least stable isomer having the highest pentagon adjacency index (N_P =20) and the highest resistance distance (R_T =955.15). In Table 1 we summarized the computed values of pentagon arm signatures $\{n_0, n_1, n_2, n_3, n_4, n_5\}$, the pentagon adjacency indices N_P and the topological descriptors Ψ, for the forty C_{40} isomers. (Each of isomers is labeled according Fowler and Manolopoulos [1]). Simultaneously, using Density Functional Tight-Binding (DFTB) method [16] we calculated the total energy values Q_C characterizing the relative stability of isomers.

Table 1
Topological parameters and relative energies of the forty C_{40} isomers

Isomer	Topological parameters								Energy,Q_C (eV)
	n_0	n_1	n_2	n_3	n_4	n_5	N_P	Ψ	
C40:38	0	8	0	4	0	0	10	0.8140	-342,031
C40:39	0	10	0	0	0	2	10	0.8106	-341,631
C40:31	1	3	5	3	0	0	11	0.7631	-341,438
C40:29	2	2	4	4	0	0	11	0.7628	-341,345
C40:26	2	6	2	2	0	0	11	0.7108	-341,094
C40:24	3	4	3	2	0	0	11	0.7102	-341,022
C40:37	4	6	0	2	0	0	11	0.6744	-340,636
C40:40	0	0	12	0	0	0	12	0.6924	-340,580
C40:14	3	2	5	2	0	0	12	0.6715	-340,476
C40:36	4	6	2	0	0	0	11	0.6597	-340,431
C40:30	3	3	3	3	0	0	12	0.6711	-340,304
C40:25	4	4	2	2	0	0	12	0.6382	-340,277
C40:22	5	3	3	1	0	0	12	0.6219	-340,230
C40:35	4	6	2	0	0	0	11	0.6597	-340,196
C40:21	6	2	0	4	0	0	12	0.6358	-340,151
C40:27	4	6	0	2	0	0	12	0.6219	-340,126
C40:15	2	8	2	0	0	0	12	0.6250	-339,943
C40:17	2	6	4	0	0	0	13	0.5943	-339,884
C40:34	5	6	1	0	0	0	12	0.5923	-339,827
C40:28	4	5	2	0	0	1	12	0.6358	-339,777
C40:16	2	6	4	0	0	0	13	0.5943	-339,645
C40:20	6	6	0	0	0	0	12	0.5772	-339,627
C40:9	4	2	4	2	0	0	13	0.6075	-339,614
C40:10	6	2	4	0	0	0	13	0.5622	-339,558
C40:12	4	6	2	0	0	0	13	0.5641	-339,370
C40:13	7	2	3	0	0	0	13	0.5467	-339,347
C40:19	4	2	6	0	0	0	13	0.5933	-339,292
C40:23	8	2	2	0	0	0	13	0.5313	-338,690
C40:6	7	4	1	0	0	0	14	0.4970	-338,624
C40:18	6	6	0	0	0	0	14	0.4987	-338,341
C40:5	6	1	4	0	0	1	14	0.5497	-338,332
C40:32	8	4	0	0	0	0	14	0.4843	-338,270
C40:8	6	4	2	0	0	0	15	0.4785	-338,113
C40:33	4	8	0	0	0	0	14	0.5132	-337,922
C40:4	7	4	1	0	0	0	15	0.4654	-337,348
C40:7	6	6	0	0	0	0	15	0.4670	-337,330
C40:11	10	2	0	0	0	0	15	0.4404	-336,642
C40:2	8	4	0	0	0	0	16	0.4262	-336,489
C40:3	12	0	0	0	0	0	18	0.3659	-335,193
C40:1	12	0	0	0	0	0	20	0.3297	-333,806

These energies are also given in Table 1. As shown in Table 1, using the topological descriptor Ψ we get the following trends of relative stability: C40:38 > C40:39 > C40:31> C40:29. This corresponds to the theoretical results based on ab initio calculations [3-5]. This finding confirms that topological descriptor Ψ correlates highly with the computed total energy value Q_C. Moreover, from Table 1 it can be seen that in the set of C_{40} fullerenes, there are three balanced isomers: C40:40 is 2-balanced, while C40:1 and C40:3 are 0-balanced isomers.

4 Comparative Tests Performed on a Set of C_{66} Isomers

In order to test the discriminating power of topological descriptor Ψ, we used the sets of C_{66} isomers. The number of topologically different C_{66} isomers is 4478. All of them were generated and sorted in terms of the calculated total energy values. Among C_{66} fullerenes there are 3 isomers with lowest pentagon adjacency index 2, and 26 isomers with N_P =3.

Table 2
Topological parameters and relative energies of the forty lowest energy C_{66} isomers

Isomer	Topological parameters								Energy,Q_C
	n_0	n_1	n_2	n_3	n_4	n_5	N_P	Ψ	(eV)
C66:4169	2	1	0	3	4	2	2	3.4756	-583.0067
C66:4348	0	4	0	4	2	2	2	3.4214	-582.8916
C66:4466	2	0	2	2	6	0	2	3.4214	-582.7047
C66:4007	2	2	2	3	2	1	3	2.4369	-582.3229
C66:3764	2	1	5	3	1	0	3	2.3537	-582.3027
C66:4456	2	2	2	6	0	0	3	2.3537	-582.1878
C66:4462	1	2	5	4	0	0	3	2.3557	-582.1816
C66:4060	2	3	2	3	2	0	3	2.3479	-582.1267
C66:4141	1	3	2	3	3	0	3	2.4423	-582.1118
C66:4312	0	3	2	6	0	1	3	2.4871	-582.0754
C66:4439	1	3	5	3	0	0	3	2.3126	-582.0316
C66:3765	2	1	5	1	3	0	3	2.3954	-582.0278
C66:3538	2	1	4	4	0	1	3	2.3954	-582.022
C66:4447	2	1	3	5	1	0	3	2.3982	-581.9169
C66:4458	2	2	3	2	3	0	3	2.3937	-5819087
C66:4331	0	4	4	2	2	0	3	2.3995	-581.8906
C66:4454	1	4	7	0	0	0	3	2.2279	-581.8632
C66:3824	3	1	1	2	4	1	3	2.4783	-581.8594
C66:4434	2	2	3	2	3	0	3	2.3937	-581.8251
C66:4369	2	3	2	3	2	0	3	2.3479	-581.8133
C66:4388	2	1	2	3	4	0	3	2.4862	-581.8098
C66:4410	1	6	3	2	0	0	3	2.2254	-581.8034
C66:4444	1	2	6	2	1	0	3	2.3558	-581.7878
C66:4398	3	4	2	2	1	0	3	2.2054	-581.7731
C66:4409	2	3	3	3	1	0	3	2.3056	-581.7640
C66:4455	2	1	5	3	1	0	3	2.3537	-581.6897
C66:3473	0	4	0	5	2	1	3	2.5324	-581.5661
C66:4449	2	0	5	2	3	0	3	2.4423	-581.5501
C66:4433	3	5	1	3	0	0	4	1.7236	-581.4675
C66:3961	1	6	3	2	0	0	4	1.7707	-581.4670
C66:4441	2	0	4	4	2	0	3	2.4430	-581.4669
C66:4316	2	0	4	6	0	0	4	1.9179	-581.4382
C66:4297	2	3	4	3	0	0	4	1.8054	-581.3990
C66:4346	0	7	1	3	1	0	4	1.8429	-581.3902
C66:4244	3	3	5	1	0	0	4	1.7303	-581.3872
C66:4313	0	4	6	2	0	0	4	1.8427	-581.3737
C66:4430	4	2	4	2	0	0	4	1.7236	-581.3698
C66:4381	1	7	3	1	0	0	4	1.7358	-581.3404
C66:4008	2	3	2	5	0	0	4	1.8415	-581.3177
C66:4349	0	6	6	0	0	0	4	1.7698	-581.2652

Table 2 shows the pentagon arm index signature, the pentagon adjacency index N_P, the topological descriptor Ψ, and the calculated total energy values Q_C for the 40 lowest-energy isomers. According to our results, and considering the computed values of Ψ, the most stable isomer is C66:4169, while the next two isomers with minimal energies are C66:4348 and C66:4466.

In these two latter cases the topological descriptor Ψ is identical (Ψ=3.4214). The calculated energies of top 5 isomers are in agreement with the results published in Refs. [17, 18].

In ranking the isomers, due to the larger amount of information included in Ψ it was reasonable to expect that Ψ performs better than N_P. According to experiments the discriminating ability of Ψ is more efficient than that of N_P.

Summary and Conclusions

In order to characterize and classify quantitatively the local topological structure of lower fullerenes C_n with $n \leq 70$ a simple method has been suggested. The concept is based on the computation of a finite set of topological invarians called pentagon arm indices.

For stability prediction purposes, a novel three-variable topological descriptor (Ψ) has been defined. This includes not only the N_P index, but additionally two other independent topological graph invariants (M_1 and M_2) derived from the components of the pentagon arm signature vector.

To test and evaluate the discriminating power of Ψ the sets of C_{40} and C_{66} fullerene isomers have been chosen. It was demonstrated that the proposed topological descriptor Ψ is able not only to characterize the combinatorial structure of different fullerene isomers, but also to rank them in the order of decreasing stability.

Acknowledgements

This work was supported by OTKA Foundation (no. K73776) and the Hungarian National Office of Research and Technology (NKTH) as a part of a Bilateral Cooperation Program (under contract no. HR-38/2008).

References

[1] P. W. Fowler, D. E. Manolopoulos: An Atlas of Fullerenes, Calendron Press, Oxford, 1995

[2] A. T. Balaban, X. Liu, D. J. Klein, D. Babics, T. G. Schmalz, W. A. Seitz, M. Randic, Graph Invariants for Fullerenes, J. Chem. Inf. Comput. Sci., Vol. 35 (1995) pp. 396-404

[3] E. E. B. Campbell, P. W. Fowler, D. Mitchell, F. Zerbetto, Increasing Cost of Pentagon Adjacency for Larger Fullerenes, Chemical Physics Letters, Vol. 250 (1996) pp. 544-548

[4] E. Albertazzi, C. Domene et al, Pentagon Adjacency as a Determinant of Fullerene Stability, Phys. Chem. Chem. Phys., Vol. 1 (1999) pp. 2913-2918

[5] P. W. Fowler, Resistance Distances in Fullerene Graphs, Croat. Chem. Acta, Vol. 75 (2002) pp. 401-408

[6] F. Torrens, Computing the Permanent of the Adjacency Matrix for Fullerenes, Internet Electronic Journal of Molecular Design, Vol. 1 (2002) pp. 351-359

[7] S. Fajtlowitz, C. E. Larson, Graph-Theoretical Independence as a Predictor of Fullerene Stability, Chemical Physics Letters, Vol. 377 (2003) pp. 485-490

[8] P. W. Fowler, Complexity, Spanning Trees and Relative Energies, in Fullerene Isomers, MATCH Commun. Math. Comput. Chem. Vol. 48 (2003) pp. 87-96

[9] T. Réti, E. Bitay, Prediction of Fullerene Stability Using Topological Descriptors, Materials Science Forum, Vols. 537-538, (2007), pp. 439-448

[10] T. Doslic, Bipartivity of Fullerene Graphs and Fullerene Stability, Chemical Physics Letters, Vol. 412 (2005) pp. 336-340

[11] T. Doslic, Saturation Number of Fullerene Graphs, J. Math. Chem., Vol. 43, (2008), pp. 647-657

[12] M. Alcami, G. Sachez, S. Diaz-Tendero, Y. Wang, F. Martin, Structural Patterns in Fullerenes Showing Adjacent Pentagons: C_{20} to C_{72}., J. Nanosci. Nanotechnol. Vol. 7 (2007) pp. 1329-1338

[13] S. Schein, M. Sands-Kidner, A Geometric Principle May Guide Self Assembly of Fullerene Cages from Clathrin Triskella and from Carbon Atoms, Biophysical Journal, Vol. 94 (2008) pp. 958-976

[14] L. H. Gan, J. Liu, Q. Hui, S. Q. Shao, Z. H. Liu, General Geometric Rule for Stability of Carbon Polyhedra, Chemical Physics Letters, Vol. 472 (2009) pp. 224-227

[15] T. Réti, I. Lászlo, A. Graovac, Local Combinatorial Characterization of Fullerenes, in preparation

[16] D. Porezag, Th. Frauenheim, Th. Köhler, G. Seifert, R. Kaschner: Constitution of Tight-Binding-like Potentials on the Basis of Density-Functional Therory: Application to Carbon, Phys. Rev. Vol. B51 (1995) pp. 12947-12957

[17] Q. B. Yan, Q. R. Zheng, G. Su, Theoretical Study on the Structures, Properties and Spectroscopies of Fullerene Derivatives $C_{66}X_4$ (X =H, F, Cl), Carbon, Vol. 45, (2007) pp. 1821-1827

[18] J. Aihara, Kinetic Instability of Neutral and Charched Fullerenes with Four-Membered Rings, Internet Electronic J. Mol. Design, Vol. 2, (2003) pp. 492-498

Simplex Differential Evolution

Musrrat Ali[1], Millie Pant[1] and Ajith Abraham[2]

[1] Department of Paper Technology, Indian Institute of Technology Roorkee, Saharanpur campus, Saharanpur -247001, India

[2] Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, P.O. Box 2259, Auburn, Washington-98071-2259, USA

musrrat.iitr@gmail.com, millidma@gmail.com, ajith.abraham@ieee.org

Abstract: Differential evolution (DE) algorithms are commonly used metaheuristics for global optimization, but there has been very little research done on the generation of their initial population. The selection of the initial population in a population-based heuristic optimization method is important, since it affects the search for several iterations and often has an influence on the final solution. If no a priori information about the optima is available, the initial population is often selected randomly using pseudorandom numbers. In this paper, we have investigated the effect of generating the initial population without using the conventional methods like computer generated random numbers or quasi random sequences. We have applied non linear simplex method in conjugation of pseudorandom numbers to generate initial population for DE. Proposed algorithm is named as NSDE (using non linear simplex method), is tested on a set of 20 benchmark problems with box constraints, taken from literature and the numerical results are compared with results obtained by traditional DE and opposition based DE (ODE). Numerical results show that the proposed scheme considered by us for generating the random numbers significantly improves the performance of DE in terms of convergence rate and average CPU time.

Keywords: Stochastic optimization, differential evolution, crossover, initial population, random numbers

1 Introduction

DE is comparatively a recent addition to class of population based search heuristics. Nevertheless, it has emerged as one of the techniques most favored by engineers for solving continuous optimization problems. DE [1, 2] has several attractive features. Besides being an exceptionally simple evolutionary strategy, it is significantly faster and robust for solving numerical optimization problems and is more likely to find the function's true global optimum. Also, it is worth mentioning that DE has a compact structure with a small computer code and has

fewer control parameters in comparison to other evolutionary algorithms. Originally Price and Storn proposed a single strategy for DE, which they later extended to ten different strategies [3].

DE has been successfully applied to a wide range of problems including Batch Fermentation Process [4], Optimal design of heat exchanges [5], synthesis and optimization of heat integrated distillation system [6], optimization of non-linear chemical process [7], optimization of process synthesis and design problems [8], optimization of thermal cracker operation [9], optimization of water pumping system [10], dynamic optimization of a continuous polymer reactor [11], optimization of low pressure chemical vapor deposition reactors [12], etc.

Despite having several striking features and successful applications to various fields DE is sometimes criticized for its slow convergence rate for computationally expensive functions. By varying the control parameters the convergence rate of DE may be increased but it should be noted that it do not affect the quality of solution. Generally, in population based search techniques like DE an acceptable trade-off should be maintained between convergence and type of solution, which even if not a global optimal solution should be satisfactory rather than converging to a suboptimal solution which may not even be a local solution. Several attempts have been made in this direction to fortify DE with suitable mechanisms to improve its performance. Most of the studies involve the tuning or controlling of the parameters of algorithm and improving the mutation, crossover and selection mechanism, some interesting modifications that helped in enhancing the performance of DE include introduction of greedy random strategy for selection of mutant vector [13], modifications in mutation and localization in acceptance rule [14], DE with preferential crossover [15], crossover based local search method for DE [16], self adaptive differential evolution algorithm [17], new donor schemes proposed for the mutation operation of DE [18], parent centric DE [28]. All the modified versions have shown that a slight change in the structure of DE can help in improving its performance. However, the role of the initial population, which is the topic of this paper, is widely ignored. Often, the whole area of research is set aside by a statement "generate an initial population," without implying how it should be done. There is only few literature is available on this topic [23-27]. An interesting method for generating the initial population was suggested by Rahnamayan et al ([19], [20]) in which the initial population was generated using opposition based rule. To further continue the research in this direction, in this paper we propose two modified versions of DE to improve its performance in terms of convergence rate without compromising with the quality of solution. The modified version presented in this paper is named non linear simplex method called NSDE. In the present study our aim is to investigate the effect of initial population on payoff between convergence rate and solution quality. Our motivation is to encourage discussions on methods of initial population construction. Performances of the proposed algorithms are compared with Basic DE and differential evolution initialized by opposition based learning

(ODE), which is a recently modified version of differential evolution [19], on a set of twenty unconstrained benchmark problems.

Remaining of the paper is organized in following manner; in Section 2, we give a brief description of DE. In Section 3, the proposed algorithms are explained. Section 4 deals with experimental settings and parameter selection. Benchmark problems considered for the present study are given in Section 5. The performances of the proposed algorithms are compared with basic DE and ODE in Section 6. The conclusions based on the present study are finally drawn in Section 7.

2 Differential Evolution (DE)

DE starts with a population of NP candidate solutions which may be represented as $X_{i,G}$, i = 1, . . . ,NP, where i index denotes the population and G denotes the generation to which the population belongs. The working of DE depends on the manipulation and efficiency of three main operators; mutation, reproduction and selection which briefly described in this section.

Mutation: Mutation operator is the prime operator of DE and it is the implementation of this operation that makes DE different from other Evolutionary algorithms. The mutation operation of DE applies the vector differentials between the existing population members for determining both the degree and direction of perturbation applied to the individual subject of the mutation operation. The mutation process at each generation begins by randomly selecting three individuals in the population. The most often used mutation strategies implemented in the DE codes are listed below.

DE/rand/1: $V_{i,g} = X_{r_1,g} + F*(X_{r_2,g} - X_{r_3,g})$ (1a)

DE/rand/2: $V_{i,g} = X_{r_1,g} + F*(X_{r_2,g} - X_{r_3,g}) + F*(X_{r_4,g} - X_{r_5,g})$ (1b)

DE/best/1: $V_{i,g} = X_{best,g} + F*(X_{r_1,g} - X_{r_2,g})$ (1c)

DE/best/2: $V_{i,g} = X_{best,g} + F*(X_{r_1,g} - X_{r_2,g}) + F*(X_{r_3,g} - X_{r_4,g})$ (1d)

DE/rand-to-best/1:

$$V_{i,g} = X_{r_1,g} + F*(X_{best,g} - X_{r_2,g}) + F*(X_{r_3,g} - X_{r_4,g}) \quad (1e)$$

Where, i = 1, . . . , NP, r_1, r_2, $r_3 \in \{1, \ldots, NP\}$ are randomly selected and satisfy:

$r_1 \neq r_2 \neq r_3 \neq i$, $F \in [0, 1]$, F is the control parameter proposed by Storn and Price [1].

Throughout the paper we shall refer to the strategy (1a) which is apparently the most commonly used version and shall refer to it as basic version.

Crossover: once the mutation phase is complete, the crossover process is activated. The perturbed individual, $V_{i,G+1} = (v_{1,i,G+1}, \ldots, v_{n,i,G+1})$, and the current population member, $X_{i,G} = (x_{1,i,G}, \ldots, x_{n,i,G})$, are subject to the crossover operation, that finally generates the population of candidates, or "trial" vectors,$U_{i,G+1} = (u_{1,i,G+1}, \ldots, u_{n,i,G+1})$, as follows:

$$u_{j,i.G+1} = \begin{cases} v_{j,i.G+1} \; if \; rand_j \le C_r \vee j = k \\ x_{j,i.G} \qquad\qquad otherwise \end{cases} \tag{2}$$

Where, j = 1. . . n, k ∈ {1, . . . , n} is a random parameter's index, chosen once for each *i*, and the crossover rate, Cr ∈ [0, 1], the other control parameter of DE, is set by the user.

Selection: The selection scheme of DE also differs from that of other EAs. The population for the next generation is selected from the individual in current population and its corresponding trial vector according to the following rule:

$$X_{i.G+1} = \begin{cases} U_{i.G+1} \; if \; f(U_{i.G+1}) \le f(X_{i.G}) \\ X_{i.G} \qquad\qquad otherwise \end{cases} \tag{3}$$

Thus, each individual of the temporary (trial) population is compared with its counterpart in the current population. The one with the lower objective function value will survive from the tournament selection to the population of the next generation. As a result, all the individuals of the next generation are as good or better than their counterparts in the current generation. In DE trial vector is not compared against all the individuals in the current generation, but only against one individual, its counterpart, in the current generation. The pseudo code of algorithm is given here.

DE pseudo code:

Step 1: The first step is the random initialization of the parent population. Randomly generate a population of (say) NP vectors, each of n dimensions: $x_{i,j} = x_{min,j} + rand(0, 1)(x_{max,j}-x_{min,j})$, where $x_{min,j}$ and x_{max} are lower and upper bounds for j^{th} component respectively, rand(0,1) is a uniform random number between 0 and 1.

Step 2: Calculate the objective function value $f(X_i)$ for all X_i.

Step 3: Select three points from population and generate perturbed individual V_i using equation (1a).

Step 4: Recombine the each target vector x_i with perturbed individual generated in step 3 to generate a trial vector U_i using equation (2).

Step 5: Check whether each variable of the trial vector is within range. If yes, then go to step 6 else make it within range using $u_{i,j} = 2* x_{min,j} - u_{i,j}$,if $u_{i,j} < x_{min,j}$ and $u_{i,j} = 2* x_{max,j} - u_{i,j}$, if $u_{i,j} > x_{max,j}$, and go to step 6.

Step 6: Calculate the objective function value for vector U_i.

Step 7: Choose better of the two (function value at target and trial point) using equation (3) for next generation.

Step 8: Check whether convergence criterion is met if yes then stop; otherwise go to step 3.

3 Proposed NSDE Algorithm

In this section we describe the proposed NSDE algorithm and discuss the effect of embedding the proposed scheme in basic DE (as given in Section 2) on two simple benchmark examples taken from literature.

3.1 Differential Evolution with Non linear Simplex Method (NSDE)

The NSDE uses nonlinear simplex method (NSM) developed by J. A. Nelder and R. Mead [22], in conjugation with uniform random number to construct the initial population. The procedure of NSDE is outlined as follows:

1 Generate a population set P of size NP uniformly as in step 1 of DE and set k=1.

2 Generate a point by NSM Method, which has the following main operations.

2.1 Select n+ 1 point from population P randomly and evaluate function at these points.

2.2 Calculate the centroid of these points excluding the worst point, say X_{max} at which function is maximum.

2.3 *Reflection*: Reflect X_{max} through the centroid to a new point X_1. And calculate the function value at this point.

2.4 *Expansion:* If $f(X_1) <= f(X_{min})$ then perform expansion to generate a new point X_2 in the expanded region otherwise go to step 2.5. If $f(X_2) < f(X_{min})$ then include point X_2 to the population Q otherwise point X_1 is included to population Q and go to step 3.

2.5 *Contraction:* If $f(X_1) < f(X_{max})$ then produce a new point X_3 by contraction otherwise go to step 2.6. If $f(X_3) < f(X_{max})$ then add point X_3 to the population otherwise include point X_1 to the population Q and go to step 3.

2.6 *Reduction:* Reduction is performed when either of the conditions mentioned above (from step 2.3 – step 2.5) are not satisfied. In this step we have replaced the original reduction method of Nelder Mead, by generating uniformly distributed random point say X_{rand} within the specified range and include it in the population Q and go to step 3.

3 If k<NP go to step 2.1 with k = k+1 else stop.

It can be seen from the above steps that in NSDE, initially a population set P of size NP is generated uniformly to which NSM is applied NP times to generate another population set Q so as to get a total population of size 2*NP. Finally, the initial population is constructed by selecting the NP fittest points from the union of P and Q. After initialization step, NSDE algorithm works like the basic DE.

With the use of NSM method, the initial population is provided with the information of the good regions that possess each particle as a vertex of the NSM simplex in each step. The algorithm is not computationally expensive, since for each particle of the initial population one function evaluation is done, which is inevitable even if we use a randomly distributed initial population.

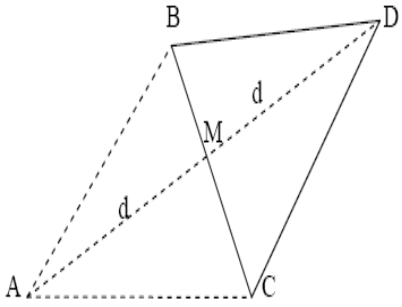


Figure 2
Reflection of A to D

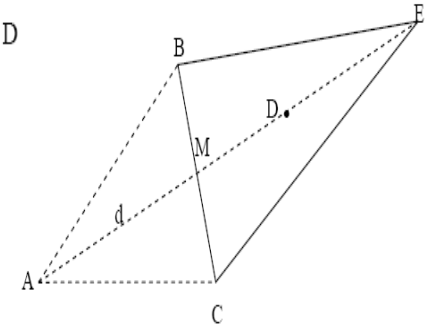


Figure 3
Expansion of D to E

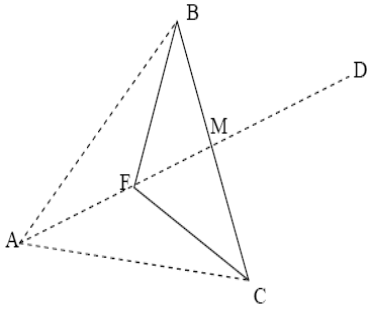


Figure 4
Contraction of D to F

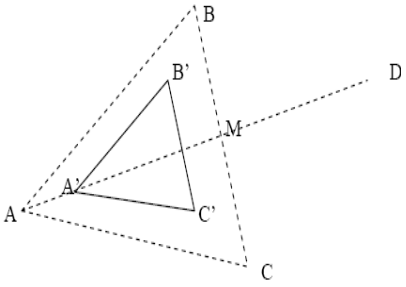


Figure 5
Reduction of ABC to A'B'C'

3.2 Effects of Using the Proposed NSDE to Generate Initial Population

The initial generation of population by nonlinear simplex method makes use of the function value to determine a candidate point for the additional population. As a result in the initial step itself we get a collection of fitter individuals which may help in increasing the efficiency of the algorithm. Consequently, the probability of obtaining the optimum in fewer NFEs increases considerably or in other words the convergence rate of the algorithm becomes faster. The initial generation of 100 points within the range [-2, 2] for Rosenbrock function and within the range [-600, 600] for Griewank function using basic DE, ODE and the proposed NSDE are depicted in Figs. 6(a)-6(c) and Figs. 7(a)-7(c) respectively. From these illustrations we can observe that the search space gets concentrated around the global optima which lies at (1, 1) with objective function value zero, for two dimensional Rosenbrock function and which lies at (0, 0) with objective function value 0, for Griewank function when the initial population is constructed using NSDE. The large search domain, [-600, 600], of Griewank function is contracted to the range of around [-400, 400] while using NSDE.

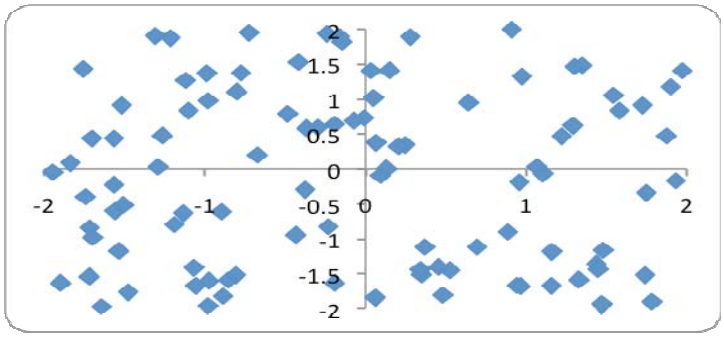


Figure 6 (a)
Initial population consisting of 100 points in the range [-2, 2] for Rosenbrock function using basic DE

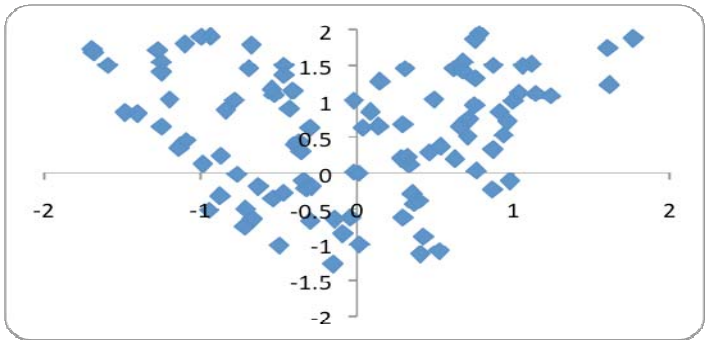


Figure 6(b)
Initial population consisting of 100 points in the range [-2, 2] for Rosenbrock function using ODE

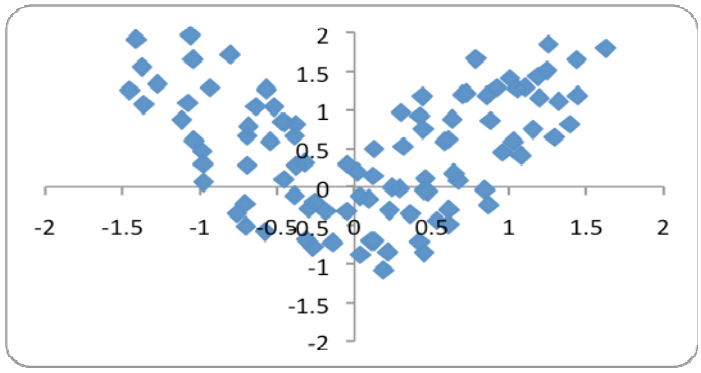


Figure 6 (c)
Initial population consisting of 100 points in the range [-2, 2] for Rosenbrock function using NSDE

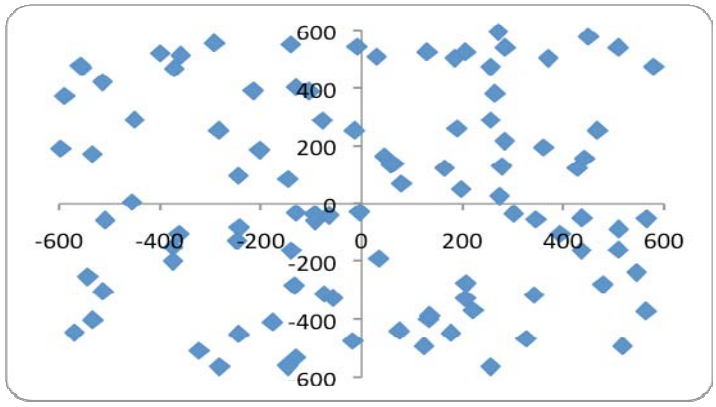


Figure 7 (a)
Initial population consisting of 100 points in the range [-600, 600] for Griewanks function using DE

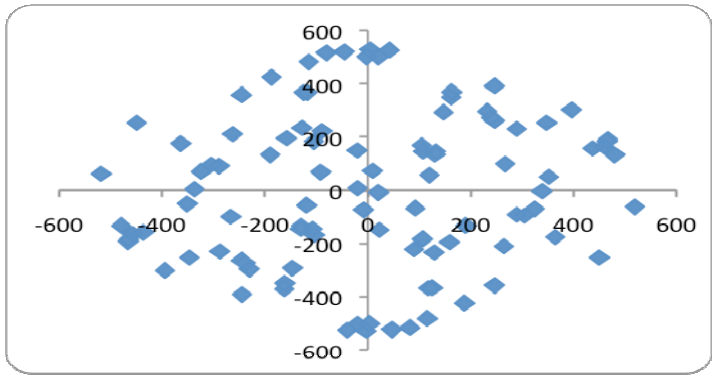


Figure 7 (b)
Initial population consisting of 100 points in the range [-600, 600] for Griewank function using ODE

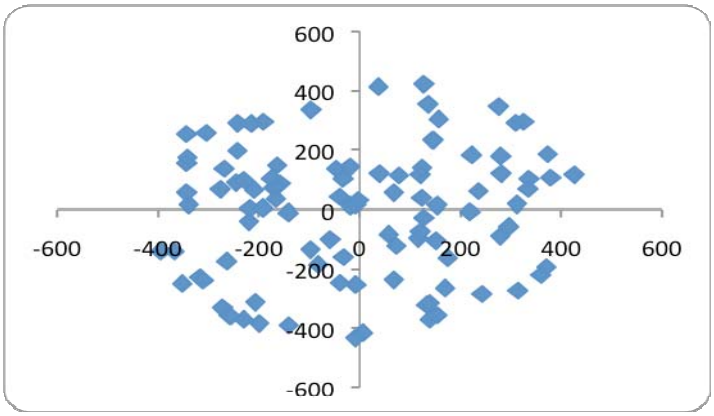


Figure 7 (c)
Initial population consisting of 100 points in the range [-600, 600] for Griewank function using NSDE

4 Experimental Setup

With DE, the lower limit for population size, NP, is 4 since the mutation process requires at least three other chromosomes for each parent. While testing the algorithms, we began by using the optimized control settings of DE. Population size, NP can always be increased to help maintain population diversity. As a general rule, an effective NP is between 3 * n and 5 * n, but can often be modified depending on the complexity of the problem. For the present study we performed several experiments with the population size as well as with the crossover rate and mutation probability rate and observed that for problems up to dimension 30 a population size of 3*n is sufficient. But here we have taken fixed population size NP=100, which is slightly large than 3*n. Values of scale F, outside the range of 0.4 to 1.2 are rarely effective, so F=0.5 is usually a good initial choice. In general higher value of C_r help in speeding up the convergence rate therefore in the present study we have taken C_r =0.9. All the algorithms are executed on a PIV PC, using DEV C++, thirty times for each problem. In order to be fair we have kept the same parameter settings for all the algorithms. Random numbers are generated using the inbuilt random number generator *rand ( ) function* available in DEVC++.

Over all acceleration rate AR, which is taken for the purpose of comparison is defined as [19]

$$AR = \left(1 - \frac{\sum_{j=1}^{\mu} NFE(\text{by one algo})_j}{\sum_{j=1}^{\mu} NFE(\text{by other algo})_j} \right) \times 100$$

Where μ is number of functions. In every case, a run was terminated when the best function value obtained is less than a threshold for the given function or when the maximum number of function evaluation (NFE=10^6) was reached. In order to have a fair comparison, these settings are kept the same for all algorithms over all benchmark functions during the simulations.

5 Benchmark Problems

The performance of proposed NSDE is evaluated on a test bed of twenty standard, benchmark problems with box constraints, taken from the literature [20]. Mathematical models of the benchmark problems along with the true optimum value are given in Appendix.

6 Numerical Results and Comparisons

6.1 Performance Comparison of Proposed NSDE with Basic DE and ODE

We have compared the proposed algorithms with the basic DE and ODE. Here we would like to mention that we have used ODE version given in [19] instead of [20] because in [20], the authors have used the additional features like opposition based generation jumping, etc. while in the present study we just focusing on the effect of initial population generation on differential evolution algorithm. Comparisons of the algorithms is done in terms of average fitness function value, standard deviation and the corresponding t-test value; average number of function evaluations and the average time taken by every algorithm to solve a particular problem. In every case, a run was terminated when the best function value obtained is less than a threshold for the given function [19] or when the maximum number of function evaluation (NFE=10^6) was reached.

From Table 1, which gives the average fitness function value, standard deviation and t-values it can be observed that for the 20 benchmark problems taken in the present study all the algorithms gave more or less similar results in terms of average fitness function value, with marginal difference, which are comparable to true optimum. For the function f_{14} (Step function) all the algorithms gives same results. The best and worst fitness function values obtained, in 30 runs, by all the algorithms for benchmark problems are given in Table 3.

However when we do the comparison in terms of average time taken and average number of function evaluations then the proposed NSDE emerges as a clear

winner. It converges to the optimum at a faster rate in comparison to all other algorithms. Only for functions f_{13} (Schwefel) NSDE took more NFE than ODE, whereas for the remaining 19 problems NSDE converged faster than the basic DE and ODE. For solving 20 problems the average NFE taken by NSDE are 1334200 while ODE took 1794738 NFE and DE took 1962733 NFE. This implies that NSDE has an acceleration rate of around 35% in comparison to basic DE and an acceleration rate of 26% in comparison to ODE. ODE on the other hand reduces the NFE only up to 8.56%, in comparison to basic DE. A similar trend of performance can be observed from the average computational time. For solving 20 problems NSDE took least CPU time in comparison to the other two algorithms.

Performance curves of selected benchmark problems are illustrated in Figs. 8(a)-8(h).

Table 1
Mean fitness, standard deviation of functions in 30 runs and t-valve

Function	Dim.	Mean fitness, (Standard deviation) and t-value		
		DE	ODE	NSDE
f_1	30	**0.0546854** (0.0131867) --	0.0901626 (0.0077778) 12.48	0.0916686 (0.00721253) 13.25
f_2	30	**0.0560517** (0.0116127) --	0.0918435 (0.00565233) 14.92	0.0866163 (0.00666531) 12.29
f_3	30	0.0957513 (0.00293408) --	0.0952397 (0.00499586) 0.48	**0.0951172** (0.00405255) 0.68
f_4	10	0.0931511 (0.0145175) --	0.0874112 (0.00699322) 1.92	**0.0851945** (0.0121355) 2.26
f_5	30	0.0915561 (0.012111) --	**0.0885065** (0.00711877) 1.17	0.0916412 (0.00860403) 0.03
f_6	30	0.0942648 (0.00478545) --	0.0933845 (0.00620528) 0.60	**0.0926704** (0.00735851) 0.98
f_7	2	**4.26112e-008** (2.5783e-008) --	6.23824e-008 (2.75612e-008) 2.82	4.9999e-008 (2.95279e-008) 1.01
f_8	4	0.0620131 (0.0239495) --	**0.0528597** (0.0276657) 1.35	0.0591064 (0.0123711) 0.58
f_9	30	0.088998 (0.00880246) --	0.092875 (0.00487147) 2.08	**0.0882776** (0.0103789) 0.29

f_{10}	10	-7.91444 (3.40729) --	-9.61563 (0.024986) 2.69	**-9.62952** (0.0238362) 2.71
f_{11}	30	**0.0842833** (0.00897659) --	0.0890837 (0.00961583) 1.97	0.0901177 (0.00969009) 2.38
f_{12}	30	0.0940407 (0.00501821) --	**0.0931232** (0.00502023) 0.69	0.0951981 (0.00373364) 0.99
f_{13}	30	0.0956696 (0.00352899) --	**0.0935369** (0.00397665) 2.16	0.0955274 (0.00495933) 0.13
f_{14}	30	0.0 (0.0) --	0.0 (0.0) --	0.0 (0.0) --
f_{15}	30	**0.0730003** (0.0169434) --	0.0880257 (0.0115251) 3.95	0.0890936 (0.00986588) 4.42
f_{16}	2	0.0645903 (0.0231492) --	0.0545825 (0.0263629) 1.54	**0.0539806** (0.0226797) 1.76
f_{17}	30	0.0910662 (0.00428958) --	**0.0845474** (0.0118228) 2.79	0.0923214 (0.00514694) 1.01
f_{18}	2	4.75455e-008 (2.9688e-008) --	3.63292e-008 (3.10335e-008) 1.41	**2.657e-008** (2.657e-008) 1.49
f_{19}	5	0.067335 (0.025448) --	**0.0738969** (0.0209749) 1.07	0.0769911 (0.0160823) 1.73
f_{20}	5	-3.99239 (0.00164918) --	**-3.99398** (0.00235545) 2.98	-3.99297 (0.00184151) 1.27

Table 2
Average CPU time (in sec) taken by the algorithms, mean number of function evaluation of 30 runs and over all acceleration rates

Fun.	Dim.	Average Time (Sec)			Mean function		
		DE	ODE	NSDE	DE	ODE	NSDE
f_1	30	0.60	0.54	**0.51**	28020	26912	**26220**
f_2	30	0.60	0.57	**0.55**	37312	36639	**35500**
f_3	30	11.30	11.10	**10.40**	295232	295112	**232860**
f_4	10	2.41	2.34	**2.13**	382454	361234	**265420**
f_5	30	1.80	1.79	**1.70**	54503	53305	**52240**
f_6	30	1.50	1.45	**1.41**	52476	51589	**49170**
f_7	2	0.31	0.29	**0.23**	3845	3740	**3520**
f_8	4	0.10	0.11	**0.09**	7902	7934	**6780**
f_9	30	1.20	1.11	1.11	44034	41455	**35330**
f_{10}	10	3.02	2.93	**2.67**	220356	196871	**172200**
f_{11}	30	2.91	2.37	**1.81**	200924	196617	**44960**
f_{12}	30	0.59	0.52	**0.50**	66154	63760	**57800**
f_{13}	30	1.83	**1.24**	1.42	197069	**148742**	155970
f_{14}	30	0.71	0.65	**0.53**	42423	41578	**32300**
f_{15}	30	0.47	0.45	**0.43**	25903	24236	**22620**
f_{16}	2	0.13	0.11	**.10**	3913	3832	**3600**
f_{17}	30	0.92	0.81	**0.76**	55029	52455	**47760**
f_{18}	2	0.23	.21	**0.16**	7367	7249	**5150**
f_{19}	5	1.66	1.12	**0.61**	205398	150173	**57540**
f_{20}	5	0.39	0.37	**0.31**	32419	31305	**27260**
Total		32.68	30.08	27.43	1962733	1794738	**1434200**
AR			7.955%	16.064%		8.5592 %	26.928 %

Table 3
Best and worst fitness function values obtained by all the algorithms

Function	Dim.	Best and Worst function values		
		DE	ODE	NSDE
f_1	30	0.0533706 0.0920816	0.0710478 0.0980475	0.0801175 0.0989173
f_2	30	0.0469366 0.0852506	0.0834493 0.0994759	0.0708503 0.0971761
f_3	30	0.0912359 0.099449	0.0812952 0.0991723	0.085954 0.0987729
f_4	10	0.0555946 0.0973456	0.0782872 0.0990834	0.0586798 0.0986525
f_5	30	0.0550155 0.0985525	0.0765341 0.0976009	0.0730851 0.0988916
f_6	30	0.0811647 0.0995538	0.0799383 0.0992613	0.0777488 0.0979521
f_7	2	3.03242e-009 8.24678e-008	1.9059e-008 9.47894e-008	5.64424e-009 8.50966e-008
f_8	4	0.0139037 0.0974824	0.00826573 0.0912189	0.0333435 0.0790444
f_9	30	0.0746445 0.0995713	0.0849655 0.098311	0.064171 0.0992847
f_{10}	10	-9.64801 -1.02642	-9.65114 -9.59249	-9.65368 -9.57805
f_{11}	30	0.0627431 0.0944119	0.0636232 0.0989899	0.0683468 0.0994605
f_{12}	30	0.0849009 0.0991914	0.0819181 0.0999306	0.0887407 0.0997806
f_{13}	30	0.0902771 0.0996024	0.0866648 0.0988438	0.0854635 0.0998667
f_{14}	30	0.0 0.0	0.0 0.0	0.0 0.0
f_{15}	30	0.0441712 0.0945989	0.0593778 0.0991945	0.067818 0.0992816
f_{16}	2	0.0277478 0.0969767	0.0129757 0.0984134	0.0297125 0.0918671
f_{17}	30	0.0844465 0.0975161	0.0593988 0.0997203	0.0836182 0.0996487
f_{18}	2	2.3063e-009 9.91416e-008	4.64862e-009 9.55725e-008	1.11418e-008 8.51751e-008
f_{19}	5	0.00746793 0.099394	0.0291998 0.0995076	0.0483536 0.0999149
f_{20}	5	-3.99626 -3.99019	-3.99737 -3.99035	-3.99644 -3.99027

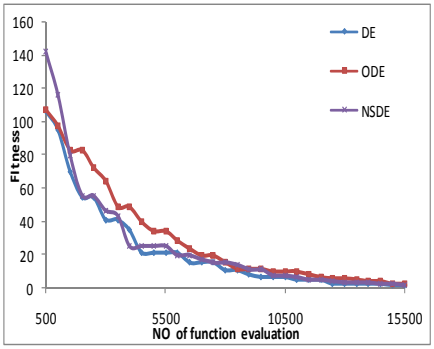


Figure 8(a)
Sphere (f_1) function

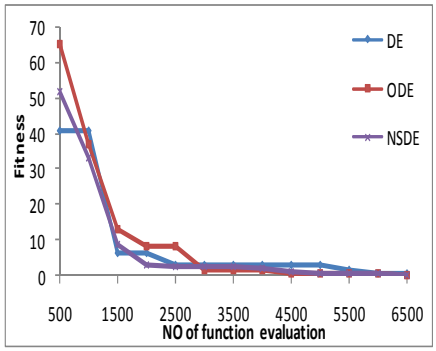


Figure 8(b)
Colville (f_8) function

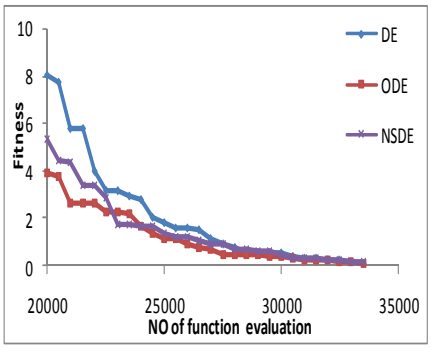


Figure 8(c)
Axis parallel (f_2) function

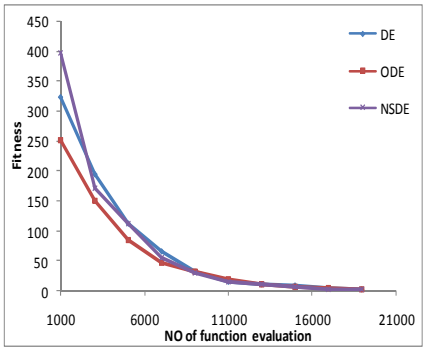


Figure 8(d)
Griewenk (f_5) function

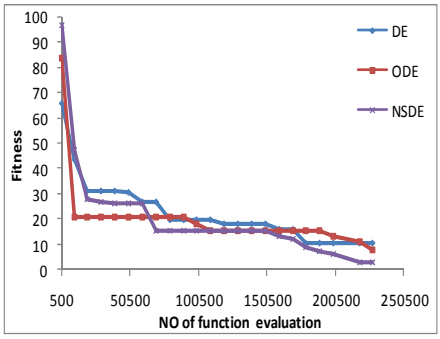


Figure 8(e)
Restrigin (f_4) function

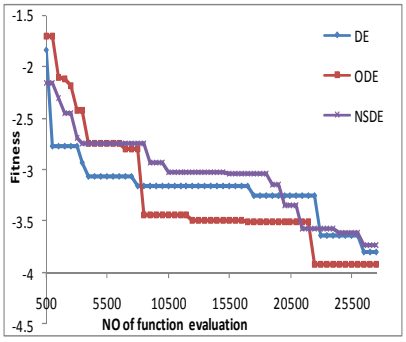


Figure 8(f)
Inverted (f_{20}) cosine

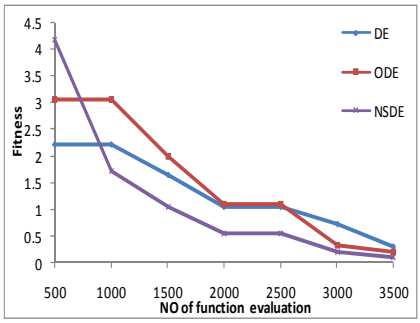


Figure 8(g)
Tripod (f_{16}) function

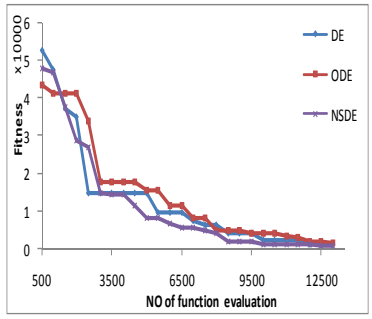


Figure 8(h)
Step (f_{14}) function

Figures 8(a)-8(h)
Performance curves of few elected benchmark problems

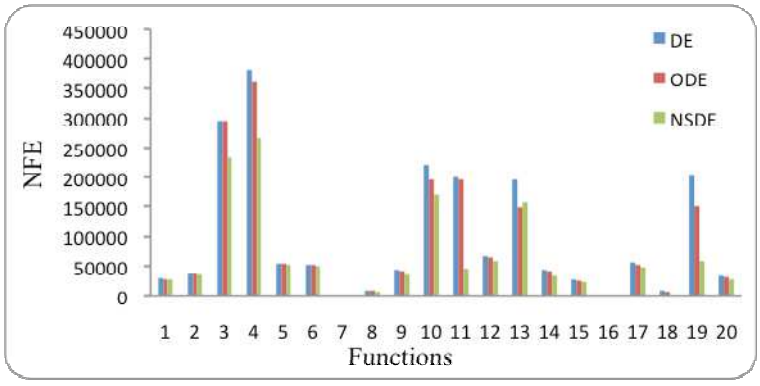


Figure 9
Average number of function evaluations taken by DE, ODE and NSDE for the 20 benchmark problems

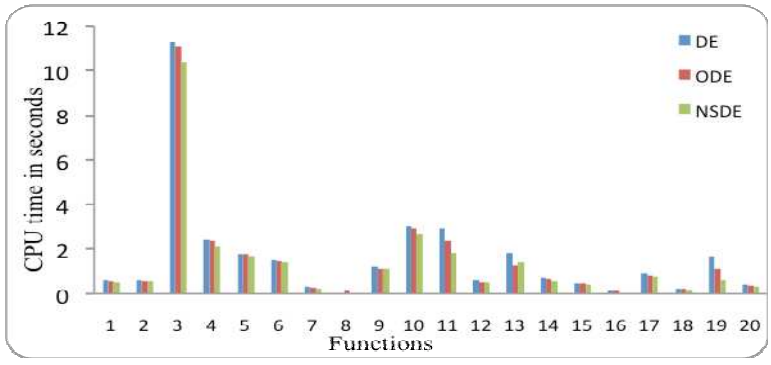


Figure 10
Average CPU time (in sec) taken by DE, ODE and NSDE for the 20 benchmark problems

Discussion and Conclusions

In the present paper we have proposed a simple and modified variant of DE namely NSDE. The only structural difference between the proposed NSDE and the basic DE lies is the generation of initial population. NSDE applies nonlinear simplex method to generate the initial population. From the empirical studies and graphic illustrations we can say that the proposed schemes enhance the working of basic DE in terms of average CPU time and NFEs without compromising with the quality of solution. Also we would like to mention that other than the process of initial population construction, we have not made use of any other additional feature/parameter in the basic structure of DE. Though we have applied the proposed schemes in basic DE, they can be applied in any evolutionary algorithm which makes use of randomly generated initial points. As a concluding statement it can be said that providing the initial population with some extra information of search space is an important help for the DE algorithm, since it may lead to faster convergence and better quality of the solutions provided by the algorithm.

References

[1] R. Storn, K. Price, "DE-a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Space", *Technical Report TR-95-012, ICSI, March 1995. Available via the Internet: ftp.icsi.berkeley.edu/pub/ techreports/ 1995/tr-95-012.ps.Z,* 1995

[2] R. Storn, K. Price, "DE-a Simple and Efficient Heuristic for Global Optimization over Continuous Space", *Journal of Global Optimization* 11(4), 1997, pp. 41-359

[3] K. Price, "An Introduction to DE", *In: Corne, D., Marco, D., Glover, F. (eds.), New Ideas in Optimization, McGraw-Hill, London (UK),* 1999, pp. 78-108

[4] F. S. Wang, W. M. Cheng, "Simultaneous Optimization of Feeding Rate and Operation Parameters for Fed-Batch Fermentation Processes", *Biotechnology Progress* 15, 1999, pp. 949-952

[5] B. V. Babu, S. A. Munawar, "Differential Evolution for the Optimal Design of Heat Exchangers", *In: Proceedings of All-India seminar on Chemical Engineering Progress on Resource Development: A Vision 2010 and Beyond, Bhuvaneshwar,* 2000

[6] B. V. Babu, R. P.Singh, "Synthesis & Optimization of Heat Integrated Distillation Systems Using Differential Evolution", *In: Proceedings of All-India seminar on Chemical Engineering Progress on Resource Development: A Vision 2010 and Beyond, Bhuvaneshwar,* 2000

[7] R Angira, B. V. Babu, "Optimization of Non-Linear Chemical Processes Using Modified Differential Evolution (MDE)", *In: Proceedings of the 2nd Indian International Conference on Artificial Intelligence, Pune, India*, 2005a, pp. 911-923

[8] R. Angira, B. V. Babu, "Optimization of Process Synthesis and Design Problems: A Modified Differential Evolution Approach", *Chemical Engineering Science* 61, 2006a, pp. 4707-4721

[9] B. V. Babu, R. Angira, "Optimization of Thermal Cracker Operation using Differential Evolution", *In: Proceedings of International Symposium & 54th Annual Session of IIChE, Chennai,* 2001b

[10] B. V. Babu, R. Angira, "Optimization of Water Pumping System Using Differential Evolution Strategies", *In: Proceedings of The Second International Conference on Computational Intelligence, Robotics, and Autonomous Systems, Singapore,* 2003

[11] M. H. Lee, C. Han, K. S. Chang, "Dynamic Optimization of a Continuous Polymer Reactor Using a Modified Differential Evolution Algorithm", *Industrial and Engineering Chemistry Research* 38, 1999, pp. 4825-4831

[12] J. C. Lu, F. S. Wang, "Optimization of Low Pressure Chemical Vapor Deposition Reactors Using Hybrid Differential Evolution", *Canadian Journal of Chemical Engineering* 79, 2001, pp. 246-254

[13] P. K Bergey, C. Rgsdale, "Modified Differential Evolution: a Greedy Random Strategy for Genetic Recombination", *omega* 33, 2005, pp. 255-265

[14] P. Kaleo, M. M. Ali, "A Numerical Study of Some Modified Differential Evolution Algorithms", *European journal of operational research* 169, 2006, pp. 1176-1184

[15] M. M. Ali, "Differential Evolution with Preferential Crossover", *European Journal of Operational Research* 181, 2007, pp. 1137-1147

[16] N. Noman, H. Iba, "Enhancing Differential Evolution Performance with Local Search for High Dimensional Function Optimization", *GECCO'05 Washington, DC, USA*, 2005, pp. 967-974

[17] A. K. Qin, P. N. Suganthan, "Self Adaptive Differential Evolution Algorithm for Numerical Optimization", *IEEE*, 2005, pp. 1785-1791

[18] H. Y. Fan, J. Lampinen, G. S. Dulikravich, "Improvements to Mutation Donor Formulation of Differential Evolution", *International Congress on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems,* EUROGEN, 2003, pp. 1-12

[19] Shahryar Rahnamayan, H. R. Tizhoosh, M. M. A. Salama, "A Novel Population Initialization Method for Accelerating Evolutionary Algorithms", *computer and applied mathematics with application* 53, 2007, pp. 1605-1614

[20] S. Rahnamayan, H. R. Tizhoosh, M. A Salman, "Opposition-based Differential Evolution", *IEEE transaction on evolutionary computation* 12(1), 2008, pp. 64-79

[21] M. M. Ali, A. Torn, "Population Set-based Global Optimization Algorithms: Some Modifications and Numerical Studies", www.ima.umn.edu/preprints/, 2003

[22] J. A. Nelder, R mead, "A Simplex Method for Function Minimization", *computer journal*, Vol. 7, 1965, pp. 308-313

[23] S. Kimura, K. Matsumura, "Genetic Algorithms using low discrepancy sequences", in proc. of GEECO 2005, pp. 1341-1346

[24] Nguyen X. H., Nguyen Q. Uy., R. I. Mckay, P. M. Tuan, "Initializing PSO with Randomized Low-Discrepancy Sequences: The Comparative Results", In Proc. of IEEE Congress on Evolutionary Algorithms, 2007, pp. 1985-1992

[25] S. Kimura, K. Matsumura, "Genetic Algorithms Using Low Discrepancy Sequences", in proceedings of the GECCO, 2005, pp. 1341-1346

[26] KE Parsopoulos, MN Vrahatis, "Initializing the Particle Swarm Optimization Using Nonlinear Simplex Method", in advances in intelligent systems, fuzzy systems, evolutionary computation, WSEAS press, 216-221, 2002

[27] Millie Pant, Radha Thangaraj, Crina Grosan, Ajith Abraham "Improved Particle Swarm Optimization with Low-Discrepancy Sequences", In Proc. of IEEE Congress on Evolutionary Algorithms, 2008, pp.

[28] Millie Pant, Musrrat Ali, V. P. Singh "Differential Evolution with Parent Centric Crossover", In Proc. of IEEE Congress on Evolutionary Algorithms, 2008, pp. 141-146

Appendix

1 Sphere function:

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \quad , \text{With} -5.12 \le x_i \le 5.12 \text{ , min } f_1(0,...,0) = 0$$

2 Axis parallel hyper-ellipsoid:

$$f_2(x) = \sum_{i=1}^{n} i x_i^2 \quad , \text{with} -5.12 \le x_i \le 5.12 \text{ , min } f_2(0,...,0) = 0$$

3 Rosenbrock's valley:

$$f_3(x) = \sum_{i=1}^{n-1} [100(x_{i+1}^2 - x_i^2) + (1 - x_i^2)] \quad \text{With} -2 \le x_i \le 2 \text{ , min } f_3(1,...,1) = 0$$

4 Restrigin's function:

$$f_4(x) = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i)) \quad \text{With} -5.12 \le x_i \le 5.12 \text{ , min}$$

$$f_4(0,...,0) = 0$$

5 Griewenk function:

$$f_5(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad \text{With} -600 \le x_i \le 600 \text{, min}$$

$$f_5(0,...,0) = 0$$

6 Ackley's function:

$$f_6(X) = -20 * \exp\left(-.2\sqrt{1/n\sum_{i=1}^{n} x_i^2}\right) - \exp\left(1/n\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e \text{ ,}$$

With $-32 \le x_i \le 32$, min $f_6(0,...,0) = 0$

7 Beale function:

$$f_7(x) = [1.5 - x_1(1-x_2)]^2 + [2.25 - x_1(1-x_2^2)]^2 + [2.625 - x_1(1-x_2^3)]^2$$

With $-4.5 \le x_i \le 4.5$, min $f_7(3, 0.5) = 0$

8 Colville function:

$$f_8(x) = 100(x_2 - x_1^2)^2 + (1-x_1)^2 + 90(x_4 - x_3^2)^2 + (1-x_3)^2 + 10.1((x_2-1)^2 + (x_4-1)^2)$$
$$19.8(x_2 - 1)(x_4 - 1)$$

,With $-10 \le x_i \le 10$, min $f_8(1,1,1,1) = 0$

9 Levy function:

$$f_9(x) = \sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)(1 + \sin^2(3\pi x_{i+1})) + (x_n - 1)(1 + \sin^2(2\pi x_n))$$

With $-10 \le x_i \le 10$, min $f_9(1,...,1) = 0$

10 Michalewicz function:

$$f_{10}(x) = -\sum_{i=1}^{n} \sin(x_i)(\sin(ix_i^2/\pi))^{2m} \quad \text{With } 0 \le x_i \le \pi \text{ , m=10, min}$$

$$f_{10(n=10)} = -9.66015$$

11 Zakharov function:

$$f_{11}(x) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5ix_i)^2 + (\sum_{i=1}^{n} 0.5ix_i)^4 \quad \text{With} -5 \le x_i \le 10 \text{, min}$$

$$f_{11}(0,...,0) = 0$$

12 Schawefel's problem 2.22:

$$f_{12}(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \quad \text{With} -10 \le x_i \le 10 \text{, min } f_{12}(0,...,0) = 0$$

13 Schwefel's problem 2.21:

$$f_{13}(x) = \max_i \{|x_i|, 1 \le i \le n\} \text{ With } -100 \le x_i \le 100 \text{, min } f_{13}(0,...,0) = 0$$

14 Step function:

$$f_{14}(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2 \text{ With } -100 \le x_i \le 100 \text{, min}$$

$$f_{14}(-0.5 \le x_i \le 0.5) = 0$$

15 Quartic function:

$$f_{15}(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1) \text{ With } -1.28 \le x_i \le 1.28 \text{, min}$$

$$f_{15}(0,...,0) = 0$$

16 Tripod function:

$$f_{16}(x) = p(x_2)(1 + p(x_1)) + |(x_1 + 50\,p(x_2)(1 - 2\,p(x_1)))| + |(x_2 + 50(1 - 2\,p(x_2)))|$$

With $-100 \le x_i \le 100$, min

$$f_{16}(0,-50) = 0 \text{ where } p(x) = 1 \text{ for } x > 0 \text{ otherwise } p(x) = 0$$

17 Alpine function:

$$f_{17}(x) = \sum_{i=1}^{n} |x_i \sin(x_i) + 0.1 x_i| \text{ With } -10 \le x_i \le 10 \text{, min } f_{17}(0,...,0) = 0$$

18 Cshaffer's function 6:

$$f_{18}(x) = 0.5 + \frac{\sin^2 \sqrt{(x_1^2 + x_2^2)} - 0.5}{1 + 0.01(x_1^2 + x_2^2)^2} \text{ With } -10 \le x_i \le 10 \text{, min}$$

$$f_{18}(0,0) = 0$$

19 Pathological function:

$$f_{19}(x) = \sum_{i=1}^{n-1} \left(0.5 + \frac{\sin^2 \sqrt{(100 x_i^2 + x_{i+1}^2)} - 0.5}{1 + 0.001(x_i^2 + x_{i+1}^2 - 2 x_i x_{i+1})^2} \right)$$

With $-100 \le x_i \le 100$, min $f_{19}(0,...,0) = 0$

20 Inverted cosine wave function:

$$f_{20}(x) = -\sum_{i=1}^{n-1} \left(\exp\left(\frac{-(x_i^2 + x_{i+1}^2 + 0.5 x_i x_{i+1})}{8}\right) \cos\left(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5 x_i x_{i+1}}\right) \right)$$

With $-5 \le x_i \le 5$, min $f_{20}(0,...,0) = -n + 1$