Landscape Analysis Under Measurement Error

Khulood Alyahya
K.Alyahya@exeter.ac.uk
University of Exeter, EX4 4QF, UK

Ozgur E. Akman
O.E.Akman@exeter.ac.uk
University of Exeter, EX4 4QF, UK

Jonathan E. Fieldsend
J.E.Fieldsend@exeter.ac.uk
University of Exeter, EX4 4QF, UK

ABSTRACT

There are situations where the need for optimisation with a global precision tolerance arises — for example, due to measurement, numerical or evaluation errors in the objective function. In such situations, a global tolerance $\epsilon > 0$ can be predefined such that two objective values are declared equal if the absolute difference between them is less than or equal to ϵ. This paper presents an overview of fitness landscape analysis under such conditions. We describe the formulation of common landscape categories in the presence of a global precision tolerance. We then proceed by discussing issues that can emerge as a result of using tolerance, such as the increase in the neutrality of the fitness landscape. To this end, we propose two methods to exhaustively explore plateaus in such application domains — one of which is point-based and the other of which is set-based.

CCS CONCEPTS

• **Theory of computation → Design and analysis of algorithms**; **Theory of randomized search heuristics**; • **Mathematics of computing → Combinatorial optimization**;

KEYWORDS

Tolerance, precision, neutrality, landscape analysis

ACM Reference Format:
Khulood Alyahya, Ozgur E. Akman, and Jonathan E. Fieldsend. 2019. Landscape Analysis Under Measurement Error. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3319619.3326858

1 INTRODUCTION

In many real world problems, the function to optimise often suffers from numerical errors due to, for example, truncation error, round-off error, or some other random variation. One such example is when the objective function is calculated from numerical approximations to the solutions of a computational model (*e.g.* a differential equation system) [1, 2, 4]. Another example is approximation error in surrogate optimisation, where obtaining high accuracy might be computationally expensive. In such situations, optimisation may be performed under a predefined global precision tolerance $\epsilon > 0$ such that two objective values, $f(\mathbf{x})$ and $f(\mathbf{y})$, are declared equal $\iff |f(\mathbf{x}) - f(\mathbf{y})| \leq \epsilon$. (Note that the tolerance studied in this paper is different to the tolerance or error threshold in [9], but is the same as that considered in [11].)

Landscape analysis offers a way of understanding optimisation problems and giving insights into the design of appropriate search heuristics. However, the presence of numerical error in the objective function can result in landscape features, such as superfluous local optima, that are merely an artefact of this numerical error. In order to avoid such misleading features being identified, the analysis of the landscape should be done in such a way as to tolerate this precision error. However, although using tolerance has the beneficial effect of smoothing *untrue* ruggedness of the landscape, it has the concomitant effect of introducing or increasing the size of neutral areas [11]. It is therefore of particular interest to characterise neutral areas in such landscapes. To this end, here we propose two alternative methods to exhaustively explore plateaus, and discuss the issues and limitation of each of them.

Understanding neutrality in the search space and its effect on the performance of search heuristics has attracted considerable attention over the past years [5, 8, 10, 12]. In [10], neutral walks were used to gain information about the neutral areas, where in each step a neutral neighbour is selected to increase the distance from the starting point. In [8], a neutrality-based iterated local search that allows neutral walks over plateaus was found to be able to find improving solutions compared with a classical iterated local search, contradicting the general belief that neutrality usually hinders local search algorithms. However, these studies relied on an error-free ($\epsilon = 0$) assumption on the observed fitness value. In reality, such optimisation problems are a special case — arguably, most practical optimisation problems are subject to at least some degree of measurement error, due to *e.g.* rounding error, sensor measurement error, estimation error, *etc.* Note that here we limit our considerations to a bounded error on the observation, as opposed to *e.g.* where experienced error is from a Gaussian distribution, where the error is effectively unbounded. There are other forms of uncertainty leading to *robust* optimisation problems, where for instance the uncertainty may be derived from tolerances in the realisation of a design (*e.g.* due to engineering precision), or through scenario sets where a design's performance is characterised over a set of fitness functions (see [7] for a discussion on this topic).

The paper proceeds as follows. In Section 2 we introduce our notation, the landscape categories commonly used for describing points with respect to local search, and introduce their analogous versions where there is an ϵ-tolerance on fitness values. In Section 3 we detail a number of algorithms to characterise neutral regions in a search landscape, in the zero error case, and the $\epsilon > 0$ case. Concluding remarks and future work are presented in Section 4.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3326858

2 NOTATION AND DEFINITIONS

Formally, a fitness landscape is a triple $(\mathcal{X}, N, f)$, where $\mathcal{X}$ is the set of candidate solutions, $N : \mathcal{X} \rightarrow 2^{\mathcal{X}}$ is a neighbourhood operator specifying the connectivity between candidate solutions and f is the fitness or objective function $f : \mathcal{X} \rightarrow \mathbb{R}$ — which is considered here, without loss of generality, to be minimised.

For a given point $\mathbf{x} \in \mathcal{X}$, according to the topology and fitness values of its direct neighbourhood, it can belong to one of seven different search position types [6]:

- Strict local minimum (SLMIN): $\forall \mathbf{y} \in N(\mathbf{x}),\ f(\mathbf{y}) > f(\mathbf{x})$.
- Non-strict local minimum (NSLMIN): $\forall \mathbf{y} \in N(\mathbf{x}),\ f(\mathbf{y}) \geq f(\mathbf{x})$, and $\exists\ \mathbf{u}, \mathbf{z} \in N(\mathbf{x})$, such that $f(\mathbf{u}) = f(\mathbf{x}) \wedge (f(\mathbf{z}) > f(\mathbf{x}))$.
- Interior plateau (IPLAT): $\forall\ \mathbf{y} \in N(\mathbf{x}),\ f(\mathbf{y}) = f(\mathbf{x})$.
- Ledge (LEDGE): $\exists\ \mathbf{u}, \mathbf{y}, \mathbf{z} \in N(\mathbf{x})$, such that $f(\mathbf{u}) = f(\mathbf{x}), f(\mathbf{y}) > f(\mathbf{x})$, and $f(\mathbf{z}) < f(\mathbf{x})$.
- Slope (SLOPE): $\forall \mathbf{y} \in N(\mathbf{x}),\ f(\mathbf{y}) \neq f(\mathbf{x})$, and $\exists\ \mathbf{u}, \mathbf{z} \in N(\mathbf{x})$, such that $f(\mathbf{u}) < f(\mathbf{x})$, and $f(\mathbf{z}) > f(\mathbf{x})$.
- Non-strict local maximum (NSLMAX): $\forall \mathbf{y} \in N(\mathbf{x}),\ f(\mathbf{y}) \leq f(\mathbf{x})$, and $\exists\ \mathbf{u}, \mathbf{z} \in N(\mathbf{x})$, such that $f(\mathbf{u}) = f(\mathbf{x}) \wedge (f(\mathbf{z}) < f(\mathbf{x}))$.
- Strict local maximum (SLMAX): $\forall \mathbf{y} \in N(\mathbf{x}),\ f(\mathbf{y}) < f(\mathbf{x})$.

Four of these types, namely NSLMIN, IPLAT, LEDGE and NSLMAX, have neutral neighbours. All of an IPLAT's neighbours are neutral, but the number of neutral neighbours varies between 1 and $|N(\mathbf{x})| - 1$ for NSLMIN and NSLMAX, and between 1 and $|N(\mathbf{x})| - 2$ for LEDGE. Following [3] we make a distinction between two different types of plateaus — closed and open:

Definition 2.1. Closed Plateau: A closed plateau is a set of connected non-strict local minima, with or without interior plateau points.

Definition 2.2. Open Plateau: An open plateau is a set of connected non-strict local minima, with or without interior plateau points and with at least one exit.

Definition 2.3. Exit: An exit is a neighbour of one or more configurations in the plateau, which shares the same objective value of the plateau but has at least one improving move. An exit can be a non-strict local maximum (minimum when maximising) or a ledge.

Distinguishing between these two types of plateau can be important in guiding the design of search algorithms. Indeed, in [8] the proposed local search method was able to exploit open plateaus to find better solutions. Note that exits are referred to in other studies as *portals*, see *e.g.* [8, 11].

In the zero error case, only changing the neighbourhood function has the effect of inducing different landscapes. However, in the case where f has some measurement error ϵ, changing ϵ has the effect of inducing different landscapes, even with f fixed. Therefore, the landscape under global tolerance is effectively a quadruple $(\mathcal{X}, N, f, \epsilon)$. Using a global tolerance allows neutral neighbours to differ by up to and including the value of ϵ. Given this, we define here the analogous search position types under a global tolerance ϵ as follows (Note that the following definitions are equivalent to the ones introduced earlier when $\epsilon = 0$):

- ϵ-SLMIN: $\forall \mathbf{y} \in N(\mathbf{x}),\ f(\mathbf{y}) - f(\mathbf{x}) > \epsilon$.
- ϵ-NSLMIN: $\forall \mathbf{y} \in N(\mathbf{x}), |f(\mathbf{y}) - f(\mathbf{x})| \leq \epsilon$ or $f(\mathbf{y}) - f(\mathbf{x}) > \epsilon$, and $\exists\ \mathbf{u}, \mathbf{z} \in N(\mathbf{x})$, such that $|f(\mathbf{u}) - f(\mathbf{x})| \leq \epsilon \wedge (f(\mathbf{z}) - f(\mathbf{x}) > \epsilon)$.
- ϵ-IPLAT: $\forall\ \mathbf{y} \in N(\mathbf{x}),\ |f(\mathbf{y}) - f(\mathbf{x})| \leq \epsilon$.
- ϵ-LEDGE: $\exists\ \mathbf{u}, \mathbf{y}, \mathbf{z} \in N(\mathbf{x})$, such that $|f(\mathbf{u}) - f(\mathbf{x})| \leq \epsilon, f(\mathbf{y}) - f(\mathbf{x}) > \epsilon$, and $f(\mathbf{x}) - f(\mathbf{z}) > \epsilon$.
- ϵ-SLOPE: $\forall \mathbf{y} \in N(\mathbf{x}),\ |f(\mathbf{y}) - f(\mathbf{x})| > \epsilon$, and $\exists\ \mathbf{u}, \mathbf{z} \in N(\mathbf{x})$, such that $f(\mathbf{x}) - f(\mathbf{u}) > \epsilon$ and $f(\mathbf{z}) - f(\mathbf{x}) > \epsilon$.
- ϵ-NSLMAX: $\forall \mathbf{y} \in N(\mathbf{x}), |f(\mathbf{y}) - f(\mathbf{x})| \leq \epsilon$ or $f(\mathbf{x}) - f(\mathbf{y}) > \epsilon$. $\exists\ \mathbf{u}, \mathbf{z} \in N(\mathbf{x})$, such that $|f(\mathbf{u}) - f(\mathbf{x})| \leq \epsilon \wedge (f(\mathbf{x}) - f(\mathbf{z}) > \epsilon)$.
- ϵ-SLMAX: $\forall \mathbf{y} \in N(\mathbf{x}),\ f(\mathbf{x}) - f(\mathbf{y}) > \epsilon$.

Algorithm 1 Exhaustive Plateau Exploration

```
start with x, where x is a NSLMIN or IPLAT
c ← f(x)                                   ▷ Plateau's objective value
U ← {x}                                    ▷ Set of unvisited plateau configurations
P_O ← ∅                                    ▷ Set of visited non-strict optima
P_I ← ∅                                    ▷ Set of visited interior plateau points
P_E ← ∅                                    ▷ Set of visited exit configurations
B_e ← ∅                                    ▷ Set of entry points to the plateau
B_d ← ∅                                    ▷ Set of departure points from the plateau
repeat
    Choose y ∈ U
    U ← U/{y}                              ▷ Remove from unvisited
    Exit ← false
    Entry ← false
    for all z ∈ N(y) | z ∉ P_E ∪ P_O ∪ P_I ∪ B_d ∪ B_e do
        if f(z) = c then
            U ← U ∪ {z}                    ▷ z is on the plateau
        else if f(z) < c then
            B_d ← B_d ∪ {z}                ▷ z is a departure point
            Exit ← true
        else
            B_e ← B_e ∪ {z}                ▷ z is an entry point
            Entry ← true
        end if
    end for
    if Exit then
        P_E ← P_E ∪ {y}
    else if Entry then
        P_O ← P_O ∪ {y}
    else
        P_I ← P_I ∪ {y}
    end if
until U = ∅
return (P_O, P_I, P_E, B_e, B_d)
```

3 CHARACTERISING PLATEAUS

We now describe three algorithms to characterise plateaus in a given landscape through exhaustive exploration. We start by describing an algorithm (listed in Algorithm 1) that explores plateaus when no tolerance is defined (*i.e.* when $\epsilon = 0$). The algorithm starts exploring a plateau when a NSLMIN or an IPLAT configuration is reached and keeps track of the points residing on the plateau in three different sets: non-strict optima (P_O), IPLAT points (P_I) and exits (P_E). The

Algorithm 2 Point-Based Exhaustive ϵ-Plateau Exploration

start with $\mathbf{x}$, where $\mathbf{x}$ is a ϵ-NSLMIN or ϵ-IPLAT
$U \leftarrow \{\mathbf{x}\}$ ▷ Set of unvisited plateau configurations
$P_O \leftarrow \emptyset$ ▷ Set of visited non-strict optima
$P_I \leftarrow \emptyset$ ▷ Set of visited interior plateau points
$P_E \leftarrow \emptyset$ ▷ Set of visited exits configurations
$B_e \leftarrow \emptyset$ ▷ Set of entry points to the plateau
$B_d \leftarrow \emptyset$ ▷ Set of departure points from the plateau
repeat
  Choose $\mathbf{y} \in U$
  $U \leftarrow U/\{\mathbf{y}\}$ ▷ Remove from unvisited
  Exit $\leftarrow$ false
  Entry $\leftarrow$ false
  for all $\mathbf{z} \in N(\mathbf{y}) \mid \mathbf{z} \notin P_E \cup P_O \cup P_I \cup B_d \cup B_e$ **do**
    if $|f(\mathbf{z}) - f(\mathbf{y})| \leq \epsilon$ **then**
      $U \leftarrow U \cup \{\mathbf{z}\}$ ▷ $\mathbf{z}$ is on the plateau
    else if $f(\mathbf{z}) < f(\mathbf{y})$ **then**
      $B_d \leftarrow B_d \cup \{\mathbf{z}\}$ ▷ $\mathbf{z}$ is a departure point
      Exit $\leftarrow$ true
    else
      $B_e \leftarrow B_e \cup \{\mathbf{z}\}$ ▷ $\mathbf{z}$ is an entry point
      Entry $\leftarrow$ true
    end if
  end for
  if Exit **then**
    $P_E \leftarrow P_E \cup \{\mathbf{y}\}$
  else if Entry **then**
    $P_O \leftarrow P_O \cup \{\mathbf{y}\}$
  else
    $P_I \leftarrow P_I \cup \{\mathbf{y}\}$
  end if
until $U = \emptyset$
return $(P_O, P_I, P_E, B_e, B_d)$

algorithm also keeps track of the boundary points that lead to the plateau B_e or allow a departure from the plateau B_d. Note that entry points (B_e) may lead to the plateau under first-improvement search but not necessarily under best-improvement local search. The sizes of B_d and P_E relative to the plateau size give an indication of how easy it is to escape the plateau. The sizes of P_O and P_I, alongside the size of B_e, give an indication of how likely the plateau is to be found.

When a tolerance is used, exploring a plateau becomes more complicated, as now the acceptance criterion for whether a configuration belongs to a plateau or not is dependent on the relative difference between it and the points in the plateau. A straightforward approach, that can be viewed as mimicking the behaviour of local search algorithm with plateau walks, is to consider a point-based acceptance criterion, as detailed in Algorithm 2. The problem with this approach is (a) the plateau returned by the algorithm is dependent on the entry point and the order in which the neighbours of a given point are explored, and (b) the difference between the objective values of configurations in the same plateau can exceed the predefined tolerance.

Algorithm 3 Set-Based Exhaustive ϵ-Plateau Exploration

start with $\mathbf{x}$, where $\mathbf{x}$ is a ϵ-NSLMIN or ϵ-IPLAT
$U \leftarrow \{\mathbf{x}\}$ ▷ Set of unvisited plateau configurations
$P \leftarrow \emptyset$ ▷ Set of visited plateau configurations
$B_e \leftarrow \emptyset$ ▷ Set of entry points to the plateau
$B_d \leftarrow \emptyset$ ▷ Set of departure points from the plateau
repeat
  Choose $\mathbf{y} \in U$
  $U \leftarrow U/\{\mathbf{y}\}$ ▷ Remove from unvisited
  $P \leftarrow P \cup \{\mathbf{y}\}$
  for all $\mathbf{z} \in N(\mathbf{y}) \mid \mathbf{z} \notin P \cup B_d \cup B_e$ **do**
    if $\exists \mathbf{v} \in P \cup U : |f(\mathbf{v}) - f(\mathbf{z})| \leq \epsilon$ **then**
      $U \leftarrow U \cup \{\mathbf{z}\}$ ▷ $\mathbf{z}$ is on the plateau
    else if $\exists \mathbf{v} \in P \cup U : f(\mathbf{v}) - f(\mathbf{z}) > \epsilon$ **then**
      $B_d \leftarrow B_d \cup \{\mathbf{z}\}$ ▷ $\mathbf{z}$ is a departure point
    else $\exists \mathbf{v} \in P \cup U : f(\mathbf{z}) - f(\mathbf{v}) > \epsilon$
      $B_e \leftarrow B_e \cup \{\mathbf{z}\}$ ▷ $\mathbf{z}$ is an entry point
    end if
  end for
  $U \leftarrow U \cup \{\mathbf{x} \mid \mathbf{x} \in B_d \wedge \exists(\mathbf{y} \in U \cup P) : |f(\mathbf{x}) - f(\mathbf{y})| \leq \epsilon\}$
  $P \leftarrow \{\mathbf{x} \mid \mathbf{x} \in P \wedge \nexists(\mathbf{y} \in U \cup P) : f(\mathbf{x}) - f(\mathbf{y}) > \epsilon\}$
  $U \leftarrow \{\mathbf{x} \mid \mathbf{x} \in U \wedge \nexists(\mathbf{y} \in U \cup P) : f(\mathbf{x}) - f(\mathbf{y}) > \epsilon\}$
  $B_d \leftarrow \{\mathbf{x} \mid \mathbf{x} \in B_d \wedge \exists(\mathbf{y} \in U \cup P) : f(\mathbf{y}) - f(\mathbf{x}) > \epsilon\}$
  $B_e \leftarrow \{\mathbf{x} \mid \mathbf{x} \in B_e \wedge \exists(\mathbf{y} \in U \cup P) f(\mathbf{x}) - f(\mathbf{y}) > \epsilon\}$
until $U = \emptyset$
return (P, B_e, B_d)

To address the latter issue, we propose exploring the plateau in a set-based fashion. This strategy is detailed in Algorithm 3, where the acceptance criterion is now dependent on the relative difference between the new point and the set of points visited so far (the archive). Since the entire history is considered in this case, it is more difficult to keep track of the different types of point that comprise the plateau, *i.e.* non-strict optima, ϵ-IPLAT points and exits, as this would require keeping track of the neighbours of each configuration, and updating the sets P_O, P_I and P_E accordingly. For simplicity, we therefore keep track of all the points in the plateau in a single set, which we refer to as P. We note that this set-based approach is still dependent on the starting point and the order in which the neighbours of a given point are explored. Furthermore, as this approach involves steps to remove configurations from plateaus that are not ϵ apart, it may return a disjoint set. That is, one where it is not possible to reach, via a walk using neighbourhood steps, all members of the returned set from an arbitrary set member.

4 CONCLUSIONS

In this paper, we have introduced a strategy for performing landscape analysis under a global precision tolerance $\epsilon > 0$. We formally described the analogous formulation of common landscape categories under such a tolerance. We then focused on characterising the neutral areas of the resulting landscapes, since the use of a global precision tolerance has the effect of increasing the size and/or number of these areas. Next, we presented a number of algorithms to characterise neutral regions in fitness landscapes, in both the zero error case, and the $\epsilon > 0$ case. For the $\epsilon > 0$ case, we proposed

two algorithms to characterise plateaus: one point-based and the other set-based. An issue we found when exploring neutral areas under ϵ-tolerance is the dependence on the starting point of the plateau exploration, which therefore results in a stochastic view of the landscape. As future work, we intend to perform a comparative study of the two algorithms to characterise plateaus for combinatorial and (discretised) continuous problems under different levels of global precision tolerance.

The proposed algorithms explore plateaus exhaustively; however, exhaustive methods quickly become intractable as the problem size increases. We will therefore also extend our methods to sample the neutral areas with some confidence bounds on the size and number of sampled neutral areas.

ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council [grant numbers EP/N017846/1, EP/N014391/1].

REFERENCES

[1] Ozgur E. Akman, David A. Rand, Paul E. Brown, and Andrew J. Millar. 2010. Robustness from flexibility in the fungal circadian clock. *BMC Systems Biology* 4, 1 (24 Jun 2010), 88. https://doi.org/10.1186/1752-0509-4-88

[2] Ozgur E. Akman, Steven Watterson, Andrew Parton, Nigel Binns, Andrew J. Millar, and Peter Ghazal. 2012. Digital clocks: simple Boolean models can quantitatively describe circadian systems. *Journal of The Royal Society Interface* 9, 74 (2012), 2365–2382. https://doi.org/10.1098/rsif.2012.0080 arXiv:https://royalsocietypublishing.org/doi/pdf/10.1098/rsif.2012.0080

[3] Khulood Alyahya and Jonathan E. Rowe. 2019. Landscape Analysis of a Class of NP-Hard Binary Packing Problems. *Evolutionary Computation* 27, 1 (2019), 47–73. https://doi.org/10.1162/evco_a_00237 PMID: 30365387.

[4] Kevin Doherty, Khulood Alyahya, Ozgur E. Akman, and Jonathan E. Fieldsend. 2017. Optimisation and Landscape Analysis of Computational Biology Models: A Case Study. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '17)*. ACM, New York, NY, USA, 1644–1651. https://doi.org/10.1145/3067695.3084609

[5] Edgar Galván-López and Riccardo Poli. 2006. An Empirical Investigation of How and Why Neutrality Affects Evolutionary Search. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO '06)*. ACM, New York, NY, USA, 1149–1156. https://doi.org/10.1145/1143997.1144180

[6] Holger H Hoos and Thomas Stützle. 2004. *Stochastic local search: Foundations and applications*. Elsevier.

[7] Yaochu Jin and Jürgen Branke. 2005. Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on evolutionary computation* 9, 3 (2005), 303–317.

[8] Marie-Eléonore Marmion, Clarisse Dhaenens, Laetitia Jourdan, Arnaud Liefooghe, and Sébastien Verel. 2011. NILS: A Neutrality-Based Iterated Local Search and Its Application to Flowshop Scheduling. In *Evolutionary Computation in Combinatorial Optimization*, Peter Merz and Jin-Kao Hao (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 191–202.

[9] Gabriela Ochoa. 2006. Error Thresholds in Genetic Algorithms. *Evolutionary Computation* 14, 2 (2006), 157–182. https://doi.org/10.1162/evco.2006.14.2.157 arXiv:https://doi.org/10.1162/evco.2006.14.2.157

[10] Christian M. Reidys and Peter F. Stadler. 2001. Neutrality in fitness landscapes. *Appl. Math. Comput.* 117, 2 (2001), 321 – 350. https://doi.org/10.1016/S0096-3003(99)00166-6

[11] Sébastien Verel, Philippe Collard, Marco Tomassini, and Leonardo Vanneschi. 2007. Fitness landscape of the cellular automata majority problem: View from the "Olympus". *Theoretical Computer Science* 378, 1 (2007), 54 – 77. https://doi.org/10.1016/j.tcs.2007.01.001

[12] Sébastien Verel, Gabriela Ochoa, and Marco Tomassini. 2011. Local Optima Networks of NK Landscapes With Neutrality. *IEEE Transactions on Evolutionary Computation* 15, 6 (Dec 2011), 783–797. https://doi.org/10.1109/TEVC.2010.2046175