Convergence Time Analysis for the Multi-objective Counting Ones Problem

Dirk Thierens

institute of information and computing sciences, utrecht university

technical report UU-CS-2003-046

www.cs.uu.nl

Convergence Time Analysis for the Multi-objective Counting Ones Problem

Dirk Thierens

Institute of Information and Computing Sciences, Utrecht University,
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
dirk.thierens@cs.uu.nl

Abstract. We propose a multi-objective generalisation for the well known Counting Ones problem, called the Multi-objective Counting Ones (MOCO) function. It is shown that the problem has four qualitative different regions. We have constructed a convergence time model for the Simple Evolutionary Multi-objective Optimiser (SEMO) algorithm. The analysis gives insight in the convergence behaviour in each region of the MOCO problem. The model predicts a $\ell^2 \ln \ell$ running time, which is confirmed by the experimental runs.

1 Introduction

In recent years, the evolutionary algorithms community has witnessed a vast increase in the design and application of multi-objective evolutionary algorithms (MOEAs) [**?**]. The popularity of MOEAs is largely due to the fact that, contrary to more traditional search techniques, population based search algorithms have the potential to convergence to the Pareto optimal front in a single optimisation run. At the moment, a number of different state-of-the-art MOEAs can be found in the literature. All these algorithms have their own characteristics, and are typically compared to each other in an experimental way. The formal analysis of their performance is still very limited. When evaluating the performance of a MOEA, two key questions need to be answered:

– first, how quickly and reliably does the algorithm converge ?
– second, how well does it find and maintain a diverse set of solutions ?

In this paper we address the convergence issue. A discussion on the diversity issue can be found elsewhere ([**?**] [**?**]). Existing work mainly focuses on an asymptotic analysis ([**?**] [**?**] [**?**] [**?**]). A notable exception can be found in [**?**] where a running time analysis for simple MOEAs on a specific problem - the Leading Ones Trailing Zeroes (LOTZ) function - is made. This study provides key insights complimentary to the asymptotic analyses. The specific choice of problem function is of course very important. It can be argued though that the LOTZ function is less suitable for investigating the performance of recombinative, population based, evolutionary algorithms. The convergence has an outspoken domino-like behaviour ([**?**]), and the first leading ones (resp. last trailing zeroes) take over the entire population very rapidly, thus preventing any recombination operator to contribute to the search process.

In this paper we also focus on a single problem function, and build a convergence model for a simple MOEA. The problem we propose is a multi-objective generalisation of the well known Counting Ones function (also known as OneMax). The study of the Counting Ones has provided key insights in the convergence behaviour of many single-objective evolutionary algorithms ([**?**] [**?**] [**?**] [**?**] [**?**]). By introducing the Multi-objective Counting Ones (MOCO) function, we hope to obtain similar insights in the performance of different MOEAs.

The next section formally defines the MOCO function. In Section 3 we derive the convergence model for a simple MOEA. Section 4 shows an experimental validation. A discussion and possible extensions are presented in Section 5. Finally, we conclude in Section 6.

2 The Multi-objective Counting Ones Problem

The Counting Ones function $CO(X) : \{0,1\}^\ell \rightarrow \mathbb{N}$ is defined as:

$$CO(X) = CO(x_1 x_2 \ldots x_\ell) = \sum_{i=1}^{\ell} x_i.$$

We transform the Counting Ones problem into the Multi-objective Counting Ones function $MOCO(X) : \{0,1\}^\ell \rightarrow \mathbb{R}^2$ by mapping the number of bits 1 to the unit circle ($[0 \ldots \ell] \rightarrow [0 \ldots 2\pi]$), and taking the cosine and sine:

$$MOCO(X) = (\cos\theta(X), \sin\theta(X)),$$

with

$$\theta(X) = \frac{2\pi}{\ell} CO(X).$$

The multi-objective optimisation problem is specified by searching for the maximal value of both $\cos\theta(X)$ and $\sin\theta(X)$. Figure 1 shows a plot of the MOCO function. It is easy to see that there are four qualitatively different regions to discern:

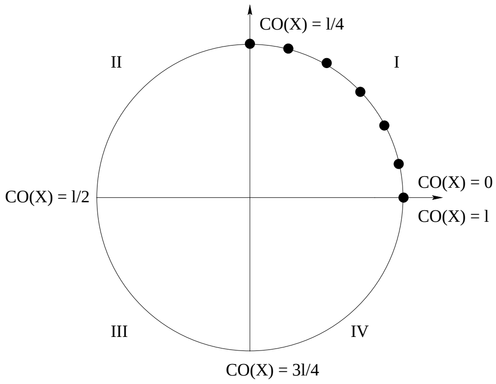

Fig. 1. The Multi-objective Counting Ones problem MOCO(X).

1. **Region I** is defined by $0 \leq \theta \leq \frac{\pi}{2}$, or equivalently, $0 \leq CO(X) \leq \frac{\ell}{4}$. None of the points on this part of the circle is Pareto dominated by any other point, therefore the entire region belongs to the Pareto optimal front.
2. **Region II** is defined by $\frac{\pi}{2} < \theta \leq \pi$, or equivalently, $\frac{\ell}{4} < CO(X) \leq \frac{\ell}{2}$. Every point in this region with $\theta = \theta_i$ Pareto dominates all points in region II and region III with $\theta = \theta_j$ and $\cos\theta_i > \cos\theta_j$. In addition it also Pareto dominates the point in region III with $\theta_j = 2\pi - \theta_i$.
3. **Region III** is defined by $\pi < \theta < \frac{3\pi}{2}$, or equivalently, $\frac{\ell}{2} < CO(X) < \frac{3\ell}{4}$. All points in this region do not Pareto dominate any other point.
4. **Region IV** is defined by $\frac{3\pi}{2} \leq \theta \leq 2\pi$, or equivalently, $\frac{3\ell}{4} \leq CO(X) \leq \ell$. Every point in this region with $\theta = \theta_i$ Pareto dominates points in region III and region IV with $\theta = \theta_j$ and $\sin\theta_j < \sin\theta_i$.

Formally, the Pareto optimal front $\mathcal{PF}$ and the corresponding Pareto optimal set $\mathcal{PS}$ are given by:

$$\mathcal{PF} = \{(\cos\theta, \sin\theta) | (0 \leq \theta \leq \frac{\pi}{2}) \vee (\theta = 2\pi)\},$$

and

$$\mathcal{PS} = \{X | (0 \leq CO(X) \leq \frac{\ell}{4}) \vee (CO(X) = \ell)\}.$$

3 Convergence model

3.1 Simple Evolutionary Multi-objective Optimiser

The single-objective counting ones problem has been extensively studied in the EA literature, and simple closed-form convergence models have been obtained. Unfortunately, the multi-objective counting ones problem does not lend itself to such a straightforward approach. The difficulty is caused by a number of factors not present in the single-objective counting ones analysis, for instance the substantially different convergence characteristics in the four regions, the influence of the archive, the elitist mechanism, the need to keep diversity, ... In addition, the diverse set of efficient Multi-objective Evolutionary Algorithm proposed in the literature (eg. NSGA, SPEA, PAES, $\mathbb{MIDEA}$), all have a different mechanism to construct the set of Pareto optimal solutions, and to encourage the search for diverse solutions. To facilitate the building of a convergence model, we restrict the analysis in this paper to a simple MOEA. The algorithm we consider here is the Simple Evolutionary Multi-objective Optimiser (SEMO) proposed by Laumanns et al. [**?**]. These authors investigated the running time complexity of SEMO on the multi-objective Leading-Ones-Trailing-Zeroes (LOTZ) problem. Although SEMO is much simpler than the full-fledged MOEAs, it possesses enough characteristics of these more complicated algorithms to make it a useful first step in the road towards a fully analytical understanding of the convergence behaviour of the state-of-the-art MOEAs. The SEMO algorithm is basically a stochastic local search algorithm, that keeps a population of all Pareto optimal solutions encountered so far. To explore the search space a randomly chosen solution from the archive is selected and mutated. Note that there is no use of recombination. Using the notation $X \succ Z$ for X Pareto dominating Z, the algorithm is specified as:

```
SEMO(ℓ)
 1  X ← GENERATE-RANDOM-BITSTRING(ℓ)
 2  P ← {X}
 3  repeat
 4          X ← RANDOMLY-CHOOSE-STRING(P)
 5          X' ← FLIP-RANDOM-BIT(X)
 6          P ← P\{Z ∈ P|X' ≻ Z}
 7          if ∄Z ∈ P such that (Z ≻ X' ∨ MOCO(Z) = MOCO(X'))
 8            then P ← P ∪ {X'}
 9    until TERMINATED-P(P)
10  return P
```

3.2 Convergence Time Analysis

Having defined the problem and the algorithm used, we can now proceed with the construction of the convergence model. Given the significant different behaviour in the four regions, we build the models for each region separately, ignoring for now the effects of the search process taking place in the other regions.

1. **Region I.** The exploration step in the SEMO algorithm consists of flipping exactly one randomly chosen bit. For any reasonable sized stringlength - say, $\ell \geq 20$ - the probability of starting the search process from within region I can be ignored. Therefore, the entire region I is generated in a fixed ordering: first, a string with $\frac{\ell}{4}$ bits-1 is generated, and since it Pareto dominates all points in the regions II and III, all points in the population P from those regions are removed (step 6 in the SEMO-algorithm). For the time being, we assume that no points from region IV have been added to P. Once a string with $\frac{\ell}{4} - 1$ bits-1 is generated, the Pareto set P contains 2 solutions, and further progress can only be obtained when the solution with the lowest number of bits equal to 1 is selected for exploration (step 4 in the SEMO-algorithm).

Formally, call i the number of bits-1 of the solution with the lowest number of bits-1 in the current population P. Call $Pr(i)$ the probability of an improvement - this is, generating a string with one bit-1 less. Recognising that an improvement is only possible when the solution with the lowest number of bits-1 in P is chosen, and one of its bits-1 is flipped, we obtain:

$$Pr(i) = \frac{1}{\frac{\ell}{4} - i + 1}\frac{i}{\ell}.$$

The expected number of trials needed to generate the entire region I, $E[T^I]$, is the sum of the expected number of trials to go from a string with i bits-1 to a string with $i-1$ bits-1, starting from a string with $\ell/4$ bits-1 until the all zeroes string is reached. $E[T^I]$ can thus be calculated as:

$$\begin{aligned}
E[T^I] &= \sum_{i=1}^{\frac{\ell}{4}} \frac{1}{Pr(i)} \\
&= \sum_{i=1}^{\frac{\ell}{4}} \frac{\ell}{i}(\frac{\ell}{4} - i + 1) \\
&= \ell(\frac{\ell}{4} + 1)\sum_{i=1}^{\frac{\ell}{4}} \frac{1}{i} - \frac{\ell^2}{4} \\
&\leq \ell(1 + (\frac{\ell}{4} + 1)\ln\frac{\ell}{4}) \\
&= O(\ell^2 \ln \ell)
\end{aligned}$$

The inequality is obtained by taking the upper boundary approximation for the harmonic series: $\sum_{i=1}^{n} \frac{1}{i} \leq \ln n + 1$.

2. **Region II.** Solutions in region II Pareto dominate all other solutions in this region that have a larger number of bits-1. Let us assume the search starts at a string with $\frac{\ell}{2}$ and continues into region II. Whenever a new string is generated with one bit-1 less than the current string, the new string replaces the current one in the population P. The entire region II is therefore traversed with a population size P equal to one. The probability of an improvement $Pr(i)$ when the current string in P has i bits-1 is simply:

$$Pr(i) = \frac{i}{\ell}.$$

The expected number of trials needed to traverse the entire region II, $E[T^{II}]$ is given by:

$$\begin{aligned}
E[T^{II}] &= \sum_{i=\frac{\ell}{4}-1}^{\frac{\ell}{2}} \frac{1}{Pr(i)} \\
&= \sum_{i=\frac{\ell}{4}-1}^{\frac{\ell}{2}} \frac{\ell}{i} \\
&= \sum_{i=1}^{\ell/2} \frac{\ell}{i} - \sum_{i=1}^{\ell/4-2} \frac{\ell}{i} \\
&\leq \ell \ln \frac{2\ell}{\ell - 8} \\
&\to \ell \ln 2
\end{aligned}$$

3. **Region III.** Points in region III do not dominate each other. When the search traverses the region from the string with $\frac{\ell}{2}$ bits-1 towards the string with $\frac{3\ell}{4}$ bits-1 all points are kept in the

population. The probability of improvement $Pr(i)$ is similar to the one in region I, only now the number of bits-1 needs to increase:

$$Pr(i) = \frac{1}{i - \frac{\ell}{2} + 1} \frac{\ell - i}{\ell}.$$

The expected number of trials needed to traverse the entire region III, $E[T^{III}]$, can be calculated as:

$$\begin{aligned} E[T^{III}] &= \sum_{i=\frac{\ell}{2}}^{\frac{3\ell}{4}-1} \frac{1}{Pr(i)} \\ &= \sum_{i=\frac{\ell}{2}}^{\frac{3\ell}{4}-1} \frac{\ell}{\ell - i}(i - \frac{\ell}{2} + 1) \\ &= \sum_{j=\frac{\ell}{4}+1}^{\frac{\ell}{2}} \frac{\ell}{j}(\ell - j - \frac{\ell}{2} + 1) \\ &= \ell(\frac{\ell}{2} + 1) \sum_{j=\frac{\ell}{4}+1}^{\frac{\ell}{2}} \frac{1}{j} - \ell(\frac{\ell}{4} - 1) \\ &\leq \ell(1 - \frac{\ell}{4} + (\frac{\ell}{2} + 1) \ln \frac{2\ell}{\ell + 4}) \end{aligned}$$

where we have used the substitution $l - i = j$.

4. **Region IV.** Points in region IV Pareto dominate all other points in this region that have a lower number of bits-1. The region is entered at the point with $\frac{3\ell}{4}$ bits-1 and moves towards the point with ℓ bits-1, which belongs to the overall Pareto optimal set of the MOCO problem. As in region II, region IV is traversed with a population size P equal to one. The probability of an improvement $Pr(i)$ when the current string in P has i bits-1 is:

$$Pr(i) = \frac{\ell - i}{\ell}.$$

The expected number of trials needed to traverse the entire region IV, $E[T^{IV}]$ is given by:

$$\begin{aligned} E[T^{IV}] &= \sum_{i=\frac{3\ell}{4}}^{\ell-1} \frac{1}{Pr(i)} \\ &= \sum_{i=\frac{3\ell}{4}}^{\ell-1} \frac{\ell}{\ell - i} \\ &= \sum_{j=1}^{\frac{\ell}{4}} \frac{\ell}{j} \\ &\leq \ell(\ln \frac{\ell}{4} + 1). \end{aligned}$$

Until now we have modelled the convergence of the SEMO algorithm for the MOCO problem in each region separately. For the transition from region II to region I the two running times do not influence each other: when region I is entered - this is, when the string with $\frac{\ell}{4}$ bits-1 is generated - no points from region II (and III) can be in the population P.

The interaction between region II and III however is more significant. During the first generations of the search the population P expands in both regions simultaneously as long as the string with the largest number of bits-1 remains to have a higher value for $\cos\theta$ than the string with the lowest number of bits 1. If at any time in the search process this condition is not fulfilled, all points found so far in region III are Pareto dominated by the point in region II, and the population P is reduced to this single point. Once this happens it is impossible for the search to return to region III. The expected speed at which the search expands in region II and III is the same, causing most of the points in region III that have a smaller number of bits 1 than the current rightmost point to be dominated by the point in region II. Therefore the size of the population P remains small, and the time spend in this phase of the search is small compared to the $O(\ell^2 \ln \ell)$ running time of region I.

As a first order approximation we assume that the overall running time of the SEMO algorithm on the MOCO problem can be approximated by the sum of the independent expected running times in region I and region II:

$$\begin{aligned} E[T] &\approx E[T^I] + E[T^{II}] \\ &= \ell(\frac{\ell}{4}+1)\sum_{i=1}^{\frac{\ell}{4}} \frac{1}{i} - \frac{\ell^2}{4} + \sum_{i=1}^{\ell/2} \frac{\ell}{i} - \sum_{i=1}^{\ell/4-2} \frac{\ell}{i} \\ &\leq \ell(1 + \ln 2 + (\frac{\ell}{4}+1)\ln\frac{\ell}{4}) \end{aligned}$$

3.3 Bitwise Mutation

Before we experimentally test the approximated expected running times, we consider the effect of the bitwise exploration step (step 5 in the SEMO-algorithm). The convergence model assumes a 1 bit-flip stochastic search - this is, each exploration step a single, randomly chosen bit is flipped. In the standard genetic algorithm approach every bit in the string is flipped with a small, constant probability p_m. For the Counting Ones problem it has been shown that the most difficult step is getting the last bit correct, and the optimal mutation probability for this to happen is $p_m = \frac{1}{\ell}$. The exact modelling of the constant mutation probability search is rather involved. We can however get an easy approximate model for the MOCO problem when calculating the probability of the number of bits-1 flipped in one particular string with $p_m = \frac{1}{\ell}$. The probability $Pr(k|i)$ that exactly k bits-1 are flipped when there are i bits-1 in a string of length ℓ is given by the probability that exactly k bits are flipped, times the probability that these k bits are all bits-1:

$$Pr(k|i) = \binom{\ell}{k} \frac{1}{\ell^k}(1-\frac{1}{\ell})^{l-k} \prod_{j=0}^{k-1} \frac{i-j}{l-j}.$$

It is instructive to compute the ratio between the expected waiting time for two bits 1 to be flipped simultaneously in one offspring, and in two consecutive 1 bit flip mutations:

$$\begin{aligned} \frac{\frac{1}{Pr(2|i)}}{\frac{2}{Pr(1|i)}} &= \frac{\ell\frac{1}{\ell}(1-\frac{1}{\ell})^{\ell-1}\frac{i}{\ell}}{\frac{\ell(\ell-1)}{2}\frac{1}{\ell^2}(1-\frac{1}{\ell})^{\ell-2}\frac{i}{\ell}\frac{i-1}{\ell-1}} \\ &= \frac{\ell-1}{i-1} \end{aligned}$$

Considering that most time is spend when i is close to ℓ, it is clear that the vast majority of improvements are caused by single bit-flip mutations. Recognising that the probability of a single bit-flip mutation equals $(1-\frac{1}{\ell})^{\ell-1} \rightarrow e^{-1}$, we can approximate the expected running time $E[T_{p_m}]$ for the bitwise mutation algorithm ($p_m = 1/\ell$) with:

$$E[T_{p_m}] \approx e.E[T].$$

4 Experimental Validation

We compare the analytical results of the previous section with experimental results for strings with length $\ell \in \{20, 40, 60, 80, 100\}$, averaged over 100 runs. We calculate the expected running time $E[T]$ using the exact calculation for the harmonic series, and the approximated value ($\tilde{E}[T]$). The tables show the average, standard deviation, minimum, and maximum number of trials needed to generate the entire region I. The first table shows the results for the 1 bit-flip stochastic local search, while the second shows the results for the bitwise mutation algorithm. It is clear that the convergence model represents a good approximation of the average running time.

ℓ	$E[T]$	$\tilde{E}[T]$	T_{avg}	T_{std}	T_{min}	T_{max}
20	196	227	210	125	59	666
40	924	1081	976	477	310	2969
60	2334	2701	2402	1042	818	6899
80	4507	5168	4593	1717	1902	8911
100	7498	8538	8137	3466	2171	22360

ℓ	$E[T_{p_m}]$	$\tilde{E}[T_{p_m}]$	T_{avg}	T_{std}	T_{min}	T_{max}
20	532	617	483	311	89	1695
40	2511	2938	2126	994	637	5171
60	6345	7343	5965	3566	1632	27457
80	12251	14049	11567	5642	3153	36431
100	20382	23210	18386	8201	5746	47023

5 Discussion & Extensions

The model build in this paper gives an insight in the convergence behaviour of a simple MOEA for the MOCO problem. This work can be extended in various ways:

- The SEMO algorithm spends a lot of time expanding the Pareto set in region I. Most of these trials cannot be successful since the population P can only be expanded when the point with the lowest number of bits-1 is selected for exploration. A similar problem occurred for the LOTZ problem, and there the FEMO algorithm was proposed to solve this problem. It would be interesting to build a convergence model for the FEMO algorithm on the MOCO problem. One difficulty is that the interacting search in region II and III is now more influential in comparison with the search in region I.
- We have mapped the number of bits-1 to the interval $[0 \ldots 2\pi]$. We could however also map it to the interval $[0 \ldots 4\pi]$, thus creating a discontinuous Pareto optimal front.
- The analysis has been limited to a non-recombinative MOEA. It is important that similar convergence models are constructed for recombinative, population based MOEAs.
- The MOCO problem considered has two objectives. An interesting question would be to build a convergence model for the 3-dimensional MOCO, where we map the number of bits-1 to the unit sphere as opposed to the unit circle.

6 Conclusion

We have proposed a multi-objective generalisation for the well known Counting Ones problem, called the Multi-objective Counting Ones (MOCO) function. We have constructed a convergence time model for this problem when running the Simple Evolutionary Multi-objective Optimiser (SEMO) algorithm. The analysis gives insight in the convergence behaviour in the four qualitative different regions of the problem. The model predicts a $\ell^2 \ln \ell$ running time, which is confirmed by experimental runs.