Dynamic Compartmental Models for Algorithm Analysis and Population Size Estimation

Hugo Monzón
Shinshu University
Nagano, Japan
hugo91@gmail.com

Hernán Aguirre
Shinshu University
Nagano, Japan
ahernan@shinshu-u.ac.jp

Sébastien Verel
Univ. Littoral Cote d'Opale
Calais, France
verel@lisic.univ-littoral.fr

Arnaud Liefooghe
Univ. Lille, CNRS, CRIStAL
Inria Lille – Nord Europe
Lille, France
arnaud.liefooghe@univ-lille1.fr

Bilel Derbel
Univ. Lille, CNRS, CRIStAL
Inria Lille – Nord Europe
Lille, France
bilel.derbel@univ-lille1.fr

Kiyoshi Tanaka
Shinshu University
Nagano, Japan
ktanaka@shinshu-u.ac.jp

ABSTRACT

Dynamic Compartmental Models (DCM) can be used to study the population dynamics of Multi- and Many-objective Optimization Evolutionary Algorithms (MOEAs). These models track the composition of the instantaneous population by grouping them in compartments and capture their behavior in a set of values, creating a compact representation for analysis and comparison of algorithms. Furthermore, the use of DCMs is not limited to analysis, by creating models of the same algorithm with different configurations is possible to extract new models by interpolation, and use them to explore fine-grained configurations lying between the ones used as a base. We illustrate the use of the model on some Multi- and Many-objective algorithms, run on enumerable MNK-Landscapes instances with 6 objectives for the analysis, and 5 objectives when used as a tool to do configuration.

CCS CONCEPTS

• **Theory of computation → Evolutionary algorithms**;

KEYWORDS

Empirical study, Working principles of evolutionary computing, Genetic algorithms, Multi-objective optimization, Compartmental Models, Modeling

ACM Reference Format:
Hugo Monzón, Hernán Aguirre, Sébastien Verel, Arnaud Liefooghe, Bilel Derbel, and Kiyoshi Tanaka. 2019. Dynamic Compartmental Models for Algorithm Analysis and Population Size Estimation. In *Proceedings of the Genetic and Evolutionary Computation Conference 2019 (GECCO '19).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3319619.3326912

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

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3326912

1 INTRODUCTION

In recent years Multi- and Many-objective Evolutionary Algorithms (MOEAs) have become established tools to solve optimization problems in several areas [2]. However, until now their use and development lacks a deeper understanding of their dynamics, what makes them successful sometimes, and what makes them fail. Modeling is a powerful tool that could improve this situation, capturing MOEAs properties in parameters that ease their analysis and could lead towards improvements or new algorithms. Predictions obtained could serve for algorithm selection or exploring configurations.

Dynamic Compartmental Models (DCMs) [5], are linear models inspired in epidemiological compartmental models [3]. Similarly to the way these models track the health status of individuals in a population, DCMs do the same using the non-domination status of solutions to capture the dynamics of MOEAs. In these models, it is assumed that a MOEA population can be split into two or more non-overlapping groups by some set of rules based on Pareto dominance. The population size always remain constant, while group sizes variate in a linear manner and the model tracks these changes. By doing so, given the compartments composition at any time t, the model can estimate how they will change in time $t + 1$ and so on.

The key element of the model is that translates algorithm dynamics' in terms of compartments and parameters which can be used for analysis or configuration of algorithms as we will show in this work. For analysis we can look at the relationship between compartments and the parameters, using them to draw conclusions on the algorithm behavior during different stages of the search process. For configuration, we present a methodology that allows sampling the algorithm's configuration space through the models. Each model and its parameters are linked to an algorithm and its configuration. We propose to interpolate the model's parameters to obtain models that represent configurations that lie in between the ones used as samples. To illustrate this concept, here we focus only on population size.

2 DYNAMIC COMPARTMENTAL MODELS

In this paper we will focus on a three compartmental DCM. The population is split into three non-overlapping groups or compartments. At each generation t, the values x_t, y_t and z_t represent the

proportion of the population that belongs to each compartment, while fulfilling at all times $x_t + y_t + z_t = 1$. If we were to represent the DCM in a time discrete manner, we would arrive at the following equation system.

$$\begin{cases} x_{t+1} = (1 - (\alpha + \beta))x_t + \bar{\alpha} y_t + \bar{\beta} z_t \\ y_{t+1} = \alpha x_t + (1 - (\bar{\alpha} + \gamma))y_t + \bar{\gamma} z_t \\ z_{t+1} = \beta x_t + \gamma y_t + (1 - (\bar{\beta} + \bar{\gamma}))z_t \\ x_t + y_t + z_t = 1 \end{cases} \quad (1)$$

where α and β are coefficients that represent the loss in x_t which becomes a gain for y_t and z_t, respectively. $\bar{\alpha}$ and γ represent the loss in y_t which becomes a gain for x_t and z_t, respectively. Finally, $\bar{\beta}$ and $\bar{\gamma}$ represent the loss in z_t.

In this work, we will use a DCM able to track performance by following the discovery of new Pareto Optimal solutions. The first rule separates solutions in two groups, the ones belonging to the Pareto Optimal (PO) set and the ones that do not. For the second rule, we take the first group and check when a solution appeared in the current generation. Solutions only found in this and only this generation belongs to the first compartment: PO Absolutely new (POA). Solutions that have appeared in any generation counting from the initial one to $t-1$ belong to the second compartment: PO Not new (PON). Finally, the third compartment contains all the solutions that are not part of the PO set: Not PO (NPO).

Having determined what solutions each compartment will track, we need to find a set of parameters that minimizes the difference between the values produced by the model $D_{\text{estimated}}$ and the actual measured values D_{measured} that comes from the algorithms runs. One easy way is to turn it into an optimization problem where we want to minimize $\frac{1}{n}\sum_{i=1}^{n}(D_{\text{estimated}} - D_{\text{measured}})^2$ in other words the mean square error *mse*. We solve this problem by using the Covariance Matrix Adaptation Evolutionary Strategy [4] (CMA-ES), a single-objective numerical optimizer.

3 ALGORITHMS AND TEST PROBLEM

The algorithms selected are four representatives Multi- and Many-objective optimization algorithms. The Non-dominated Sorting Genetic Algorithm-II (NSGA-II), Adaptive ϵ-Sampling ϵ-Hood (AϵSϵH), Indicator Based EA (IBEA) with the hypervolume (HV) indicator, and MOEA based on Decomposition (MOEA/D).

MNK-landscapes [1], a multi-objective test problem generator, is used in all experiments. The parameter K controls the ruggedness of the landscape, M is the number of objectives, and N the number of variables. The instances used here have K = 1 bit, N = 20 bits and M = 6, for the analysis example and M = 5 objectives for the population estimation example.

All algorithms were run 30 times, with population size 200 and given 100 generations for the 6 objective instance. As for the 5 objective instance, population size ranges from 150 to 750 in 50 increments, and the number of generations depends on the assigned budget of Function Evaluations (FE) and can be calculated as $Gen_{Max} = FE/Pop$. The maximum number of FE was set to 10000 simulating a limited budget equivalent to 1% of the whole search space (2^{20} total possible values).

4 ALGORITHM ANALYSIS WITH DCMS

4.1 Quality of the found models

Following the procedure described in Section 2, create a model for each algorithm on the problem instance with 6 objectives. We conducted a visual inspection of the model estimation against the measured data. For this, we take the compartments counts on the initial generation for each of the runs and use the model to estimate how they change from generation 1 to 100. Figure 1 to 3 shows the estimation for the POA, PON and NPO compartments in red, while in black is the measured values from the algorithms runs. From the plots we can see that the models are capturing the changes in each compartment for all algorithms, following the mean of all the runs.

4.2 Interpretation of the DCM parameters

Parameters in DCM relate two compartments, and in order to properly interpret them, we need to know that they encapsulate according to the equations. Let us focus α and consider the DCM model with the following compartments: POA, PON, NPO. Here this parameter relates the number of POA and PON solutions. If we were to expand the first two equations of the system (1).

$$\begin{aligned} \text{POA}_{t+1} &= \text{POA}_t - \alpha\text{POA}_t + \cdots \\ \text{PON}_{t+1} &= \text{PON}_t + \alpha\text{POA}_t + \cdots . \end{aligned}$$

Note that the values of POA and PON at time $t+1$ change in opposite directions, both proportionally to POA at time t by αPOA_t. Hence, a loss in one compartment becomes gain in the other. However, α can take positive or negative values. Thus, when $\alpha > 0$, the model tells us that the number of POA solutions reduces with time proportionally to its previous value and the same amount increases in PON. On the other hand, when $\alpha < 0$, the model says that the number of POA solutions increases proportionally to its previous value and the same amount reduces in PON. In the above example, $\alpha > 0$ can be interpreted as negative feedback of POA on itself, whereas $\alpha < 0$ a positive feedback of POA on itself.

As an example, we are going to concentrate on two algorithms, AϵSϵH and MOEA/D. We write the equations with the proper set of parameters for each algorithm and simplify them to show how the compartment depends on itself and the two other compartments. This allows us to understand better the overall dynamics of the DCMs. In the following, consider the equations marked with A as the ones for AϵSϵH and with M the ones for MOEA/D.

$$\begin{aligned} \text{POA}^A_{t+1} &= 0.5967\text{POA}^A_t + 0.0489\text{PON}^A_t + 0.0768\text{NPO}^A_t \\ \text{POA}^M_{t+1} &= 0.9621\text{POA}^M_t - 0.0288\text{PON}^M_t + 0.0319\text{NPO}^M_t \\ \\ \text{PON}^A_{t+1} &= 0.9287\text{PON}^A_t + 0.5305\text{POA}^A_t - 0.0061\text{NPO}^A_t \\ \text{PON}^M_{t+1} &= 0.9684\text{PON}^M_t + 0.0001\text{POA}^M_t + 0.0292\text{NPO}^M_t \\ \\ \text{NPO}^A_{t+1} &= 0.9293\text{NPO}^A_t - 0.1272\text{POA}^A_t + 0.0224\text{PON}^A_t \\ \text{NPO}^M_{t+1} &= 0.9389\text{NPO}^M_t + 0.0378\text{POA}^M_t + 0.0604\text{PON}^M_t \end{aligned}$$

In 6 objectives, even random sampling is able to find some PO solutions, so it is expected to find some of them in the initial population. According to the data, there are some PO solutions, so for

our model they will be absolutely new since it is the initial population, being them less than 1%, PON is 0 since there are no previous generations, and the remaining ones are NPO.

We notice that AεSεH seems to retain more than 53% of POA solutions as PON solutions, as indicated by the coefficient value of the POA_t component in the PON equation, while for MOEA/D this value is very small, less than 0.01%. Thus MOEA/D is dropping substantially more solutions than AεSεH. However, note also in POA equations that POA_t coefficient is 0.59 for AεSεH and 0.96 for MOEA/D, which indicates that MOEA/D finds many more Pareto absolutely new solutions than AεSεH. It is worth noting that POA solutions in this instance are found mostly from previous POA solutions, rather than from PON or NPO solutions. Note that the analysis done here through the parameters can also be done by inspecting the plots in Figures 1-3.

5 ESTIMATION OF POPULATION SIZES

5.1 Using splines to interpolate models

Our models are composed of six parameters $(\alpha, \beta, \gamma, \bar{\alpha}, \bar{\beta}, \bar{\gamma})$, which are linked to a particular combination of population size, algorithm and problem, so for different configurations we would have other set of parameters. We could then interpolate the values in the space of parameters, and discover this way other configurations, however for this work we only focus on the population size. In order to interpolate the models, we make a spline for each parameter, taking as data points the pair population size and parameter value.

The first step is selecting a range to explore and some sample population sizes on the range. We run the algorithms to generate the data to construct the models. Next, we fit the models to the data and get the parameters. With this data, we now create splines for each parameter. Finally, we use the splines to explore the range, generating models in between population sizes used as samples.

Here for our example, the problem was a 5 objective MNK-Landscape instance, and we decided to explore the range $\{150, 750\}$ with {150, 350, 550, 750} as sample points. We created then, new points every 50, exploring 3 new population sizes between the ones we actually had to run the algorithm. We pay the full cost of creating four models, and obtain nine more very cheaply.

5.2 Selecting Population Size under a Budget

To do selection, we consider the number of Accumulated PO (Acc. PO) solutions obtained by the algorithm with a budget of 10000 FE. We obtain this value by accumulating the estimated POA count from the model associated with each population size. For verification, we analyze the estimated Acc. PO at the end of the budget from fitted models and interpolated models, and compare then with the actual measured values that we obtain running the algorithm. We want to see if the models will allow us to discover which interval between the sampled population shows promise.

Figure 4 shows on the left for sample population sizes, the average number of Acc. PO solutions at the end of 10000 FE for 30 runs of the algorithm. This sample population sizes were used as knots for the spline. On the center we include a similar plot with values produced by the fitted models in red, while the ones from interpolated models are in black. Finally on the right side we have now the measured values for all population sizes in the range. Red ones represent the population sizes in the range used as knots for the spline, and the ones in blue the interpolation ones. We use this plot to check against the trends found by the models. In all plots the error bars show the 95% confidence interval for each mean.

Looking at Figure 4 left and center plots, we can visually compare the fitted model estimation to the measured values for the population sizes used as knots. We see that for all sizes except 150 the fitted model gets very close to the measured average, in the particular case of 750 is even a little higher. Nevertheless, as we will see, even with an underestimating model as a knot, the trend is still correctly replicated.

Now we shift the focus to the center and right figures, mainly the estimations of our models obtained through interpolation. We clearly see the influence of our first knot, 150, pulling down the estimations of the interpolated models between [150, 350]. Nevertheless in this section the growing trend is still correctly maintained when compared to the plot on the right with all the measured results.

Moving to the next interval of [350, 550], we find that our models estimations get closer to the measured results, and also follows the trend detecting that 400 and 500 obtain better values than 450. Finally in the last interval of [550, 750], the trend is not correctly replicated, as 700 places a bit higher than 650, which according to the right side of the figure should be below. Since the fitted model of 750 actually overestimates a little in this part, it seems to pull the interval making 700 seem a better option. However, the relative position of the knots points is still correct, as two of them place higher than 750 as it should be, and neither of the group is higher than the knot on left, 550.

Summarizing, if we were to trust only the model results, the information obtained from the interpolated models will guide us towards choosing a population size for our budget better than the one used in our sampling. If we partially trust the model, it still will guide us to a region worth exploring in detail.

6 CONCLUSION

In this work we presented some possible uses of Dynamic Compartmental Models, with an example as an analysis tool and also a small step forward towards algorithm configuration with an example that estimates population size given a budget. The key in both examples is the ability of parameters to capture the complex dynamics in a numerical way which opens many possibilities, specially for algorithm configuration.

In future works, we would like to propose new sets of rules that does not involve PO solutions, to be able to test the DCMs on real world problems and test the limits of this new tool.

REFERENCES

[1] Hernán Aguirre and Kiyoshi Tanaka. 2004. Insights on properties of multiobjective MNK-landscapes. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, Vol. 1. 196–203 Vol.1.

[2] Carlos A Coello Coello. 1999. An updated survey of evolutionary multiobjective optimization techniques: state of the art and future trends. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, Vol. 1. –13 Vol. 1.

[3] K. Godfrey. 1983. *Compartmental Models and Their Application.* Academic Press.

[4] Nikolaus Hansen and Andreas Ostermeier. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evol. Comput.* 9, 2 (jun 2001), 159–195.

[5] Hugo Monzón, Hernán E. Aguirre, Sébastien Vérel, Arnaud Liefooghe, Bilel Derbel, and Kiyoshi Tanaka. 2017. Closed state model for understanding the dynamics of MOEAs. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017, Berlin, Germany, July 15-19, 2017.* 609–616.

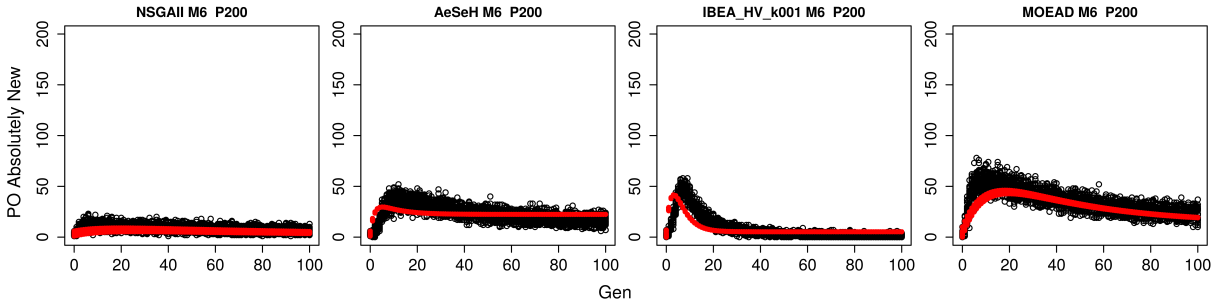

Figure 1: Model estimation vs measured PO Absolutely new data.

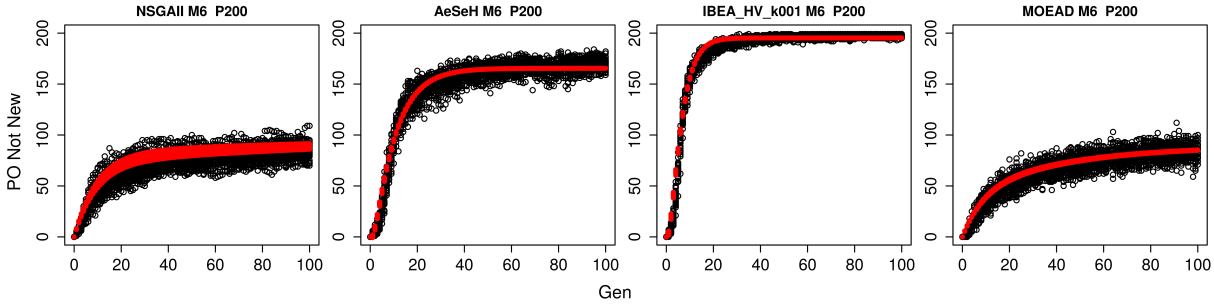

Figure 2: Model estimation vs measured PO Not new data.

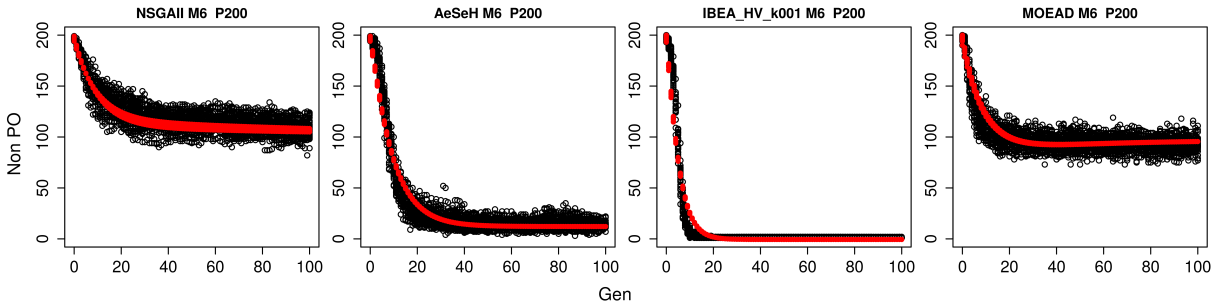

Figure 3: Model estimation vs measured Non PO data.

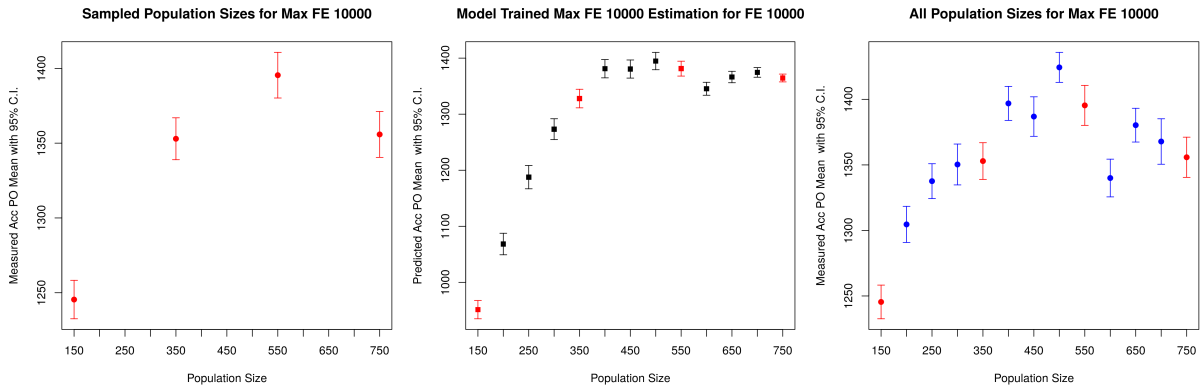

Figure 4: [Budget: 10000 FE] Average Accumulated PO over population size. Left: Sampled population sizes used as knots for the spline. Center: Model estimation for the knots and interpolated population sizes. Right: Measured values for all population sizes.