Stochastic optimization of GeantV code by use of genetic algorithms

CHEP 2016

Oksana Shadura

CERN

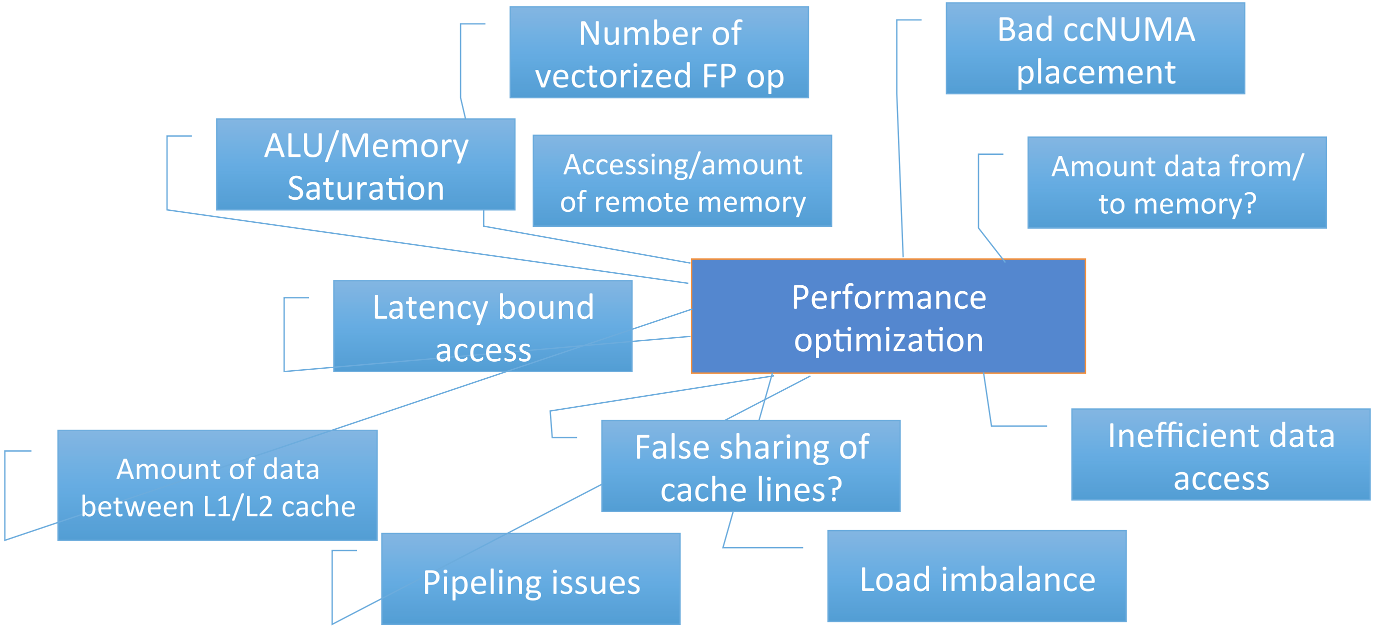
GeantV - next generation of simulation software

- GeantV is a vectorized prototype framework for the simulation of the passage of particles through matter
- Used for essential for comparisons of theory against data in High Energy Physics (HEP)
- Support different backends(Vc*, UMESIMD**) and ported to multiple modern architectures (GPU,MIC,KNL)

* https://github.com/VcDevel/Vc

** https://github.com/edanor/umesimd.git

Introduction: ideas and goals of research



How to optimize performance of HPC level?

oksana.shadura@cern.ch

Performance tuning of GeantV

Level of parallelism in GeantV
Multi-node programming
Multi-socket programming
Multi-core programing
Hardware threading: task parallelism, data parallelism
Instruction parallelism

MPI+X

Multiprocessing approach with common event queue

TBB , single-node parallelism

Vectorization

NUMA-cluster job submission

Need to be tuned on all levels

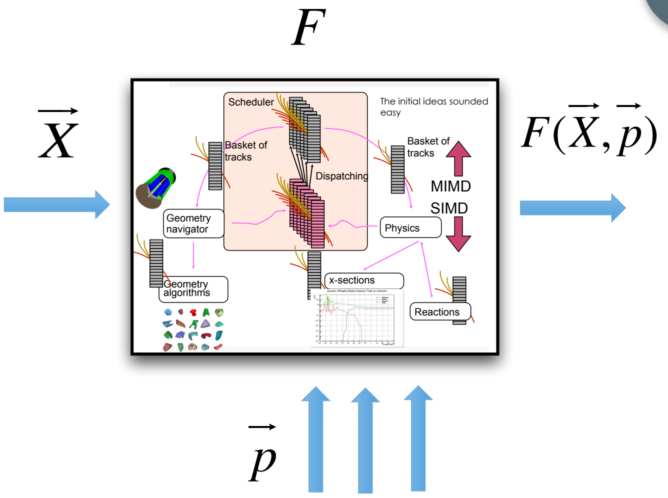
Performance tuning depends of the configuration on all levels!

GeantV as a multi-objective optimization problem (MOOP)

- A black box search task is to minimize function for which the *analytical form* is not known.

Black-box can be evaluated to obtain: $f(\vec{x}) : R^n \to R$

- **Value;**
- **Approximate gradient** (huge number of evaluations): $\frac{d\, f(x)}{d\, x} : \Delta_k f \approx \frac{f(x_k + \epsilon_k) - f(x_k)}{\epsilon_k} \Delta x_k$



External parameters, like type of computing "node"

GeantV MOOP

$$\begin{cases} \underset{x}{\text{minimize}} & f_{\text{memory}}(x) \\ \underset{x}{\text{minimize}} & f_{\text{runtime}}(x) \\ \underset{x}{\text{minimize}} & f_{\text{instructions}}(x) \\ \underset{x}{\text{maximize}} & f_{\text{nprimariespertime(x)}} \end{cases}$$

$$\begin{cases} \underset{x}{\text{minimize}} & f_{\text{cache}}(x) \\ \underset{x}{\text{minimize}} & f_{\text{etc}}(x) \\ \text{subject to} & f_{\text{memory}}(x) \leq 4000, \; i = 1, \ldots, m. \end{cases}$$

....

Definitions of decision and objective space for GeantV

Number of buffered events

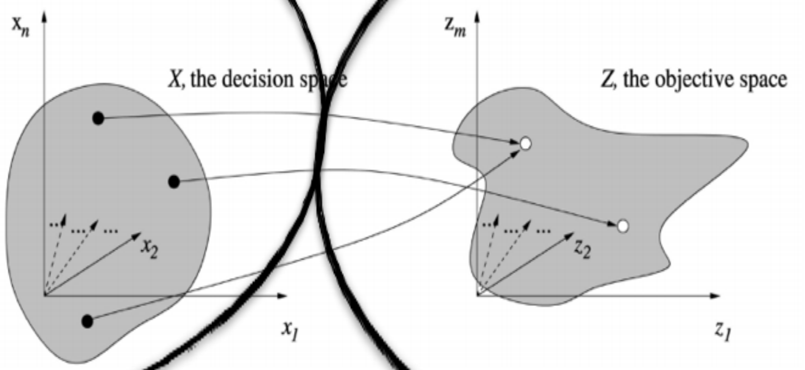
Threshold for prioritizing events

Maximum vector size

Number of steps for learning phase

Number of events

...



Number of instructions

Run Time

Number of branches

Memory

L1icache misses

L1dcache misses

Idle cycles frontend

Number of cycles

Number of primaries per RT (for simple examples)

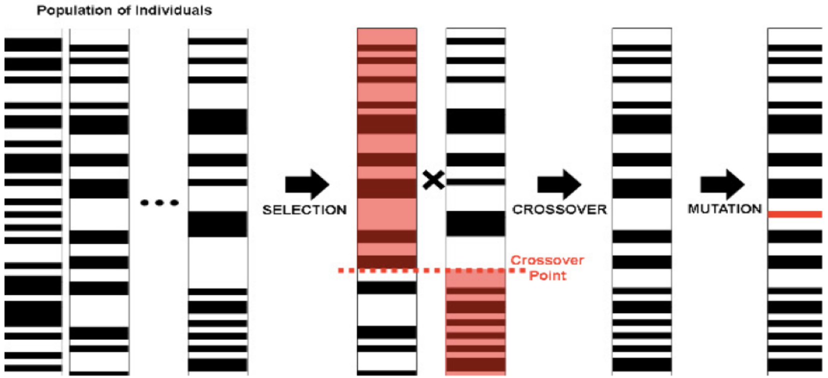
Energy efficiency

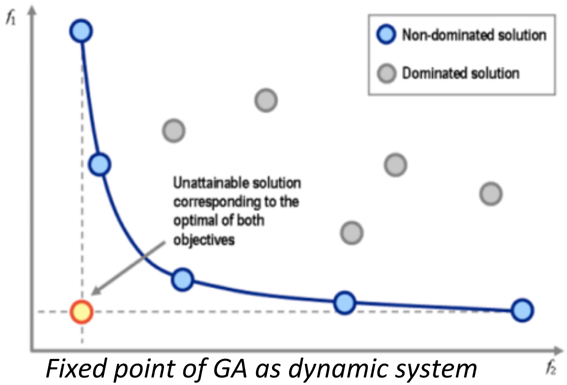
HPC parameters

Load balancing parameters

Threshold for event balancing

Genetic algorithms (GA)





Fixed point of GA as dynamic system

Selection

$F : \Lambda \to \Lambda$

$$F(\vec{p}) = \frac{\mathrm{diag}\,(f) \cdot \vec{p}}{\vec{f}^{\,t} \cdot \vec{p}}$$

Crossover

$\hat{C} : \Lambda \to \Lambda$

$$C(\vec{p}) = (\vec{p}^{t} \cdot \hat{C}_1 \cdot \vec{p}, ..., \vec{p}^{t} \cdot \hat{C}_N \cdot \vec{p})$$

GA transition matrix between population

$G : \Lambda \to \Lambda$

$$G(\vec{p}) = \hat{C} \circ \hat{U} \circ F(\vec{p})$$

$$Q_{\vec{q},\vec{p}} = \bar{M}! \prod_{\alpha \in \Omega} \frac{(G(\vec{p})_\alpha)^{\bar{M} q_\alpha}}{(\bar{M} q_\alpha)!}$$

Mutation

$\hat{U} : \Lambda \to \Lambda$

$$(\hat{U} \cdot \vec{p})_\alpha$$

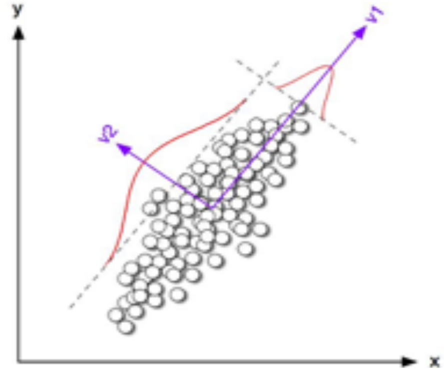
Convergence

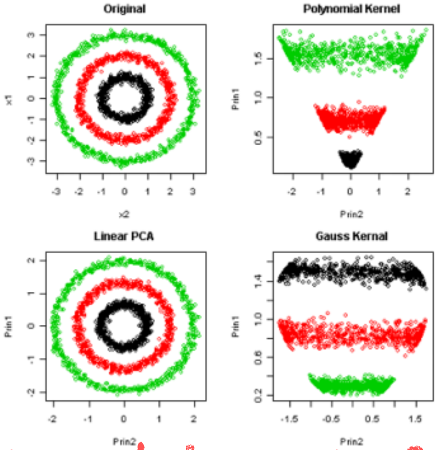
?

- GA is applicable to ***any black-box problem;***
- **No unique recipe** of type of GA for particular function;
- No needs for training or input statistics;
- ***<u>Computationally hard, but could be increased convergence rate using additional operators</u>***

Unsupervised Machine Learning: Principle Component Analysis and alternative approaches

- **Linear Principle Components Analysis (LPCA) -** captured by the **least important principal components** is noise which should be suppressed. Assuming that the variables bear a **linear relationship**, they will lie in a hyperplane and noise items will lift them away from the line. Dropping the last principal components means flattening the data in a geometric sense and eliminating some of the noise.
- **Linear Principle Components Analysis (KPCA) -** transform existing dataset to another high-dimensional space and then perform PCA on the data in that space.





Can we increase convergence by cleaning data noise?

Noise cleanup GA operator (based on idea of uncentered PCA)

Convergence genetic operator

$$G_P(\vec{p}) = \hat{P} \circ \hat{C} \circ \hat{U} \circ F(\vec{p})$$

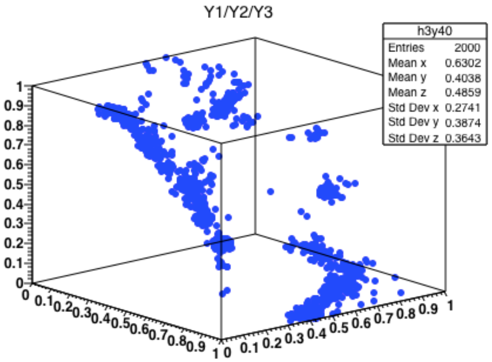
$$P: \quad V^{(u)}_{\alpha,j} = X^{(u)}_{\alpha,i} W_{i,j}, \quad W^{\,t}_{i,i'} W_{i',j} = \delta_{i,j},$$

$$\tilde{X}^{(u)}_{\alpha,i} = \sqrt{M}\left(\left[\lambda^{(u)}_1\right]^{\frac{1}{2}} \bar{V}^{(u)}_{\alpha,1} W^{\,t}_{1,i} + \ldots + \left[\lambda^{(u)}_p\right]^{\frac{1}{2}} \bar{V}^{(u)}_{\alpha,p} W^{\,t}_{p,i}\right).$$

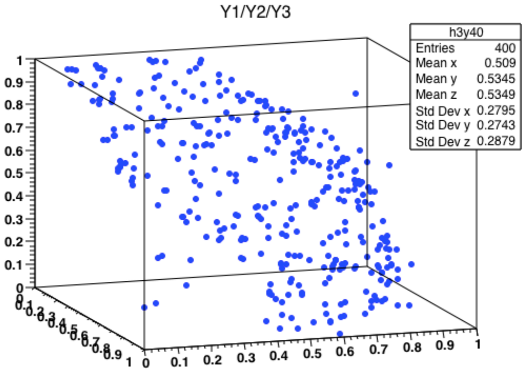
$$\eta_p = \frac{1}{Mn} \sum_{\alpha=1}^{M} \sum_{i=1}^{n} (X^{(u)}_{\alpha,i} - \tilde{X}^{(u)}_{\alpha,i})^2 = \frac{1}{n} \sum_{k=p+1}^{n} \lambda^{(u)}_k$$

(Check backup)

NSGA



NSGA-PCA

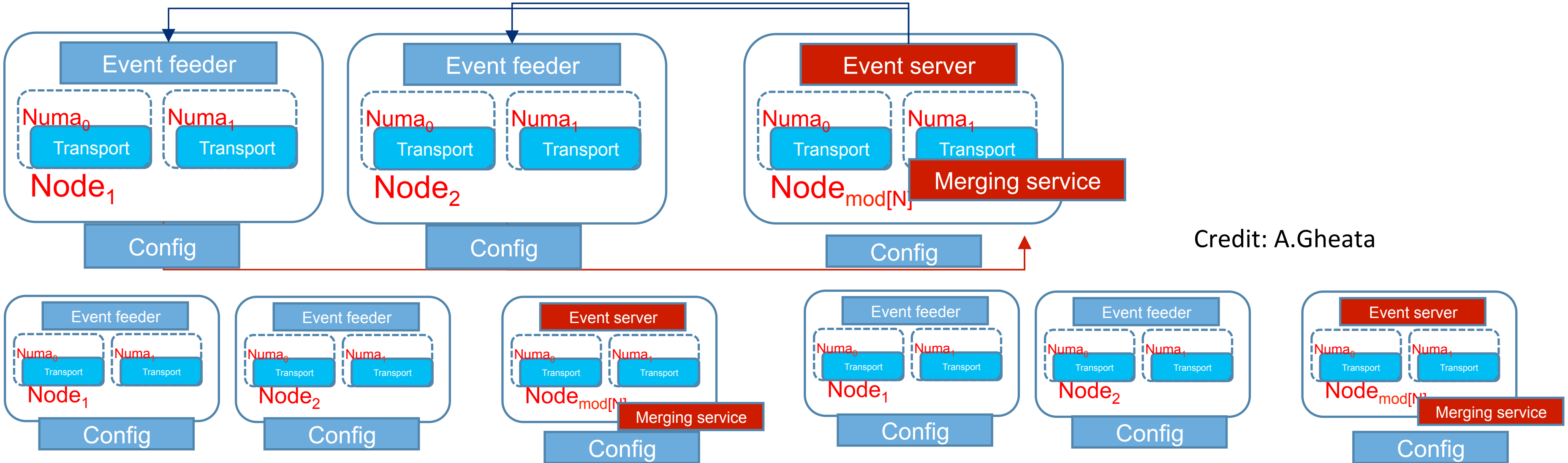


Work will be presented on IEEE Symposium on Computational Intelligence 2016

oksana.shadura@cern.ch

Tuning GeantV as a part of optimization for embarrassingly parallel simulations

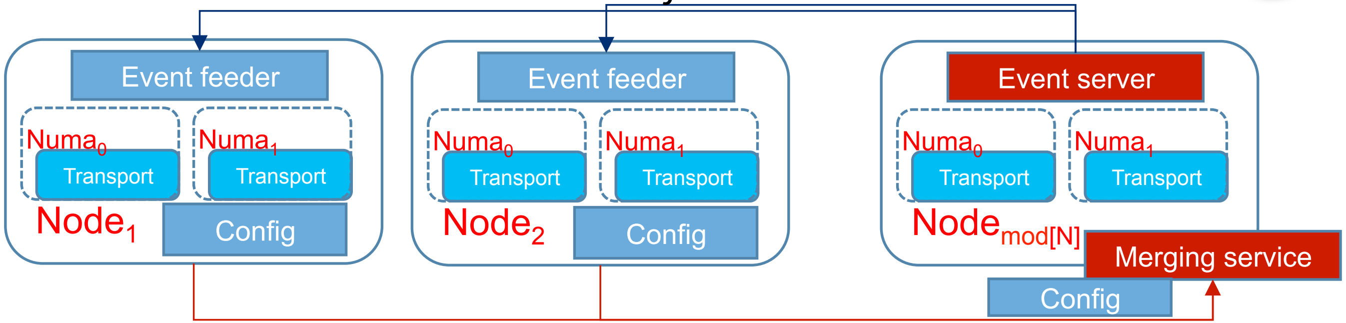
- Large set of parameters and multiple layers of parallelism creates a complex system to be tuned;
- Introduction of ***event server*** will provide stable balancing for event processing in distributed system;
  - **Event server –** concurrent server of distribution of events;
  - **Event feeder** – Event generator.
- **Configuration management** can provide **a tuning policy for HPC layer** of GeantV.

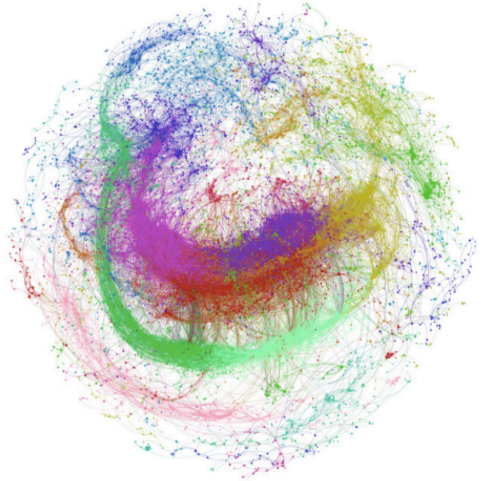


Credit: A.Gheata

Multi-layers graphs and GeantV HPC layers representation

% events between feeders?

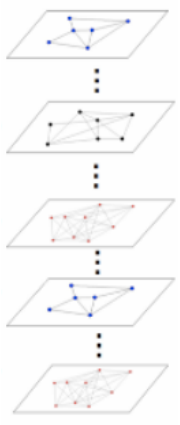




Activity in the Higgs dataset

arXiv:1504.08153 [cs.SI]

Can we create the same representation for HPC layer?



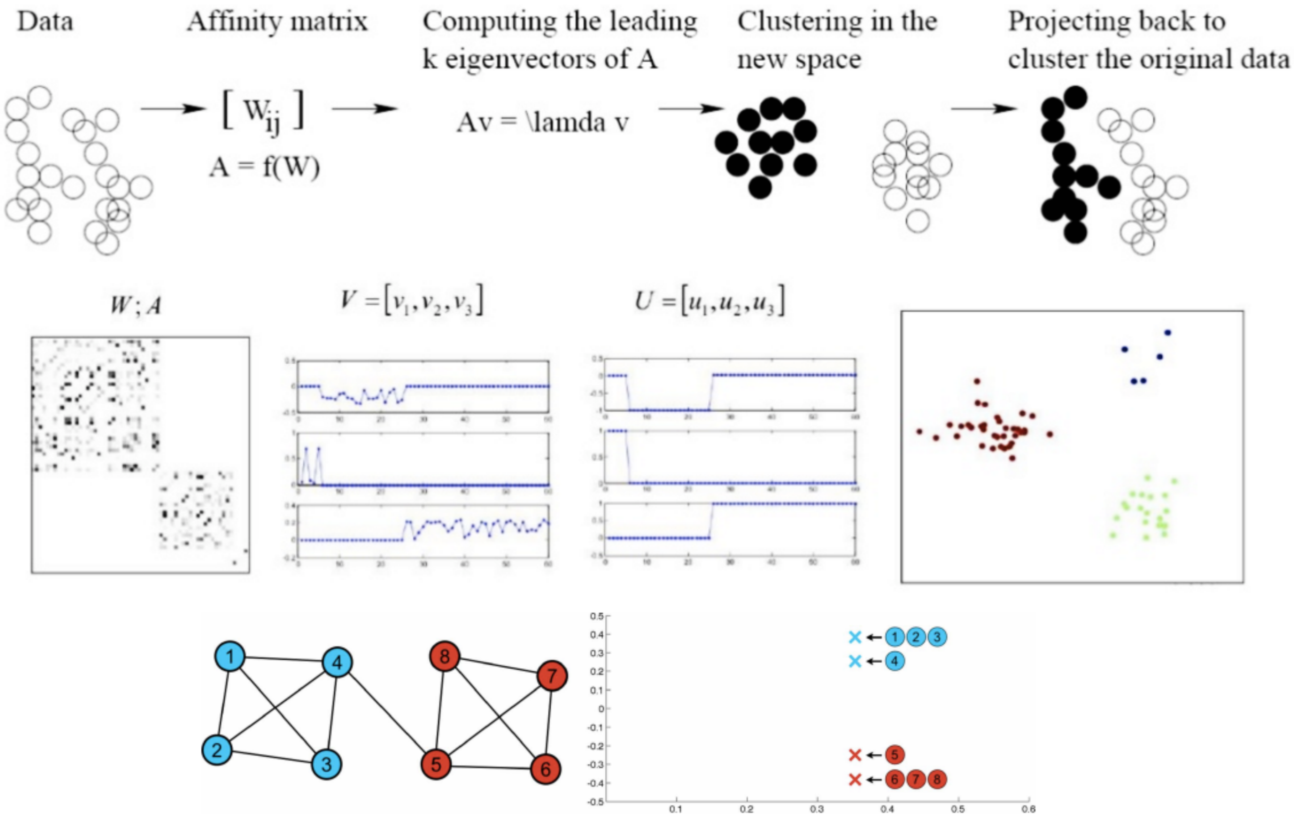
Types of physical nodes

Processors, cores

NUMA nodes (NUMA clustering)

....

Spectral clustering on multi-layer graphs

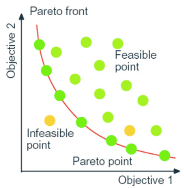


Clustering With Multi-Layer Graphs: A Spectral Perspective, *Xiaowen Dong, Pascal Frossard, Senior Member, IEEE, Pierre Vandergheynst, and Nikolai Nefedov*

Use cases for GeantV tuning

GA

Node$_1$ Node$_1$

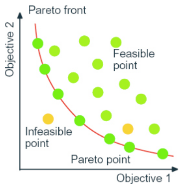


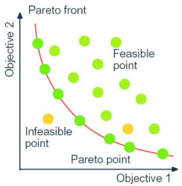
1 general Pareto front

GA1 *GA2*

Node$_1$ Node$_2$ Node$_2$

(Manually divided or based on locality detection library hwloc)



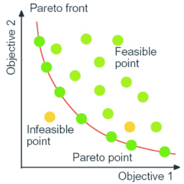


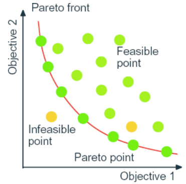
2 separate Pareto fronts

GA1 *GA1*

Node$_1$ Node$_1$ Node$_2$ Node$_2$

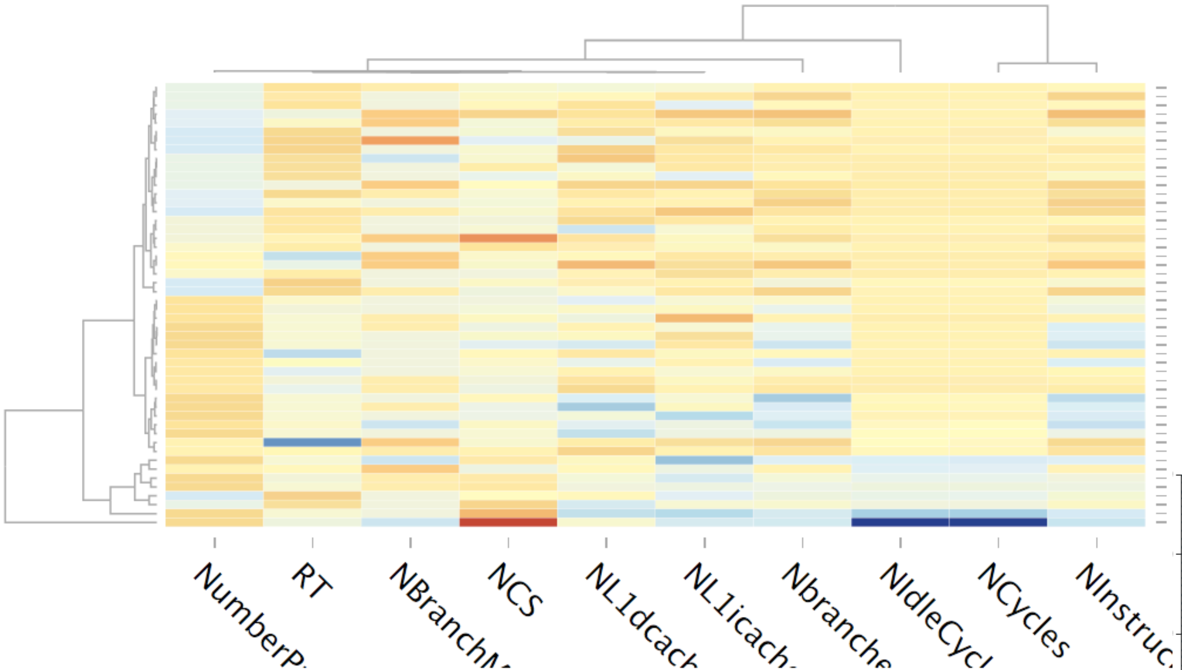
(Spectral clustering based on multi-layer graphs)

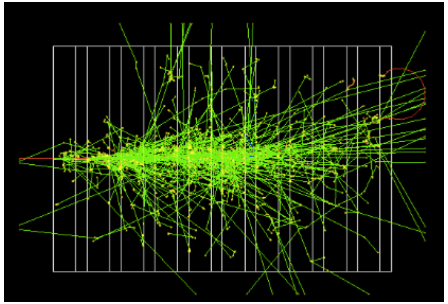


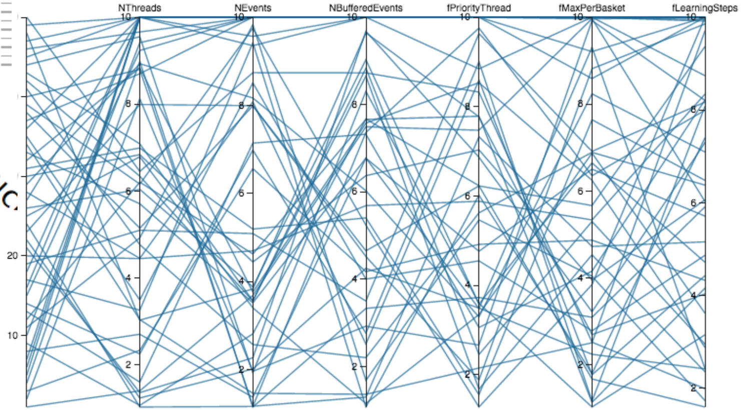


2 separate Pareto fronts

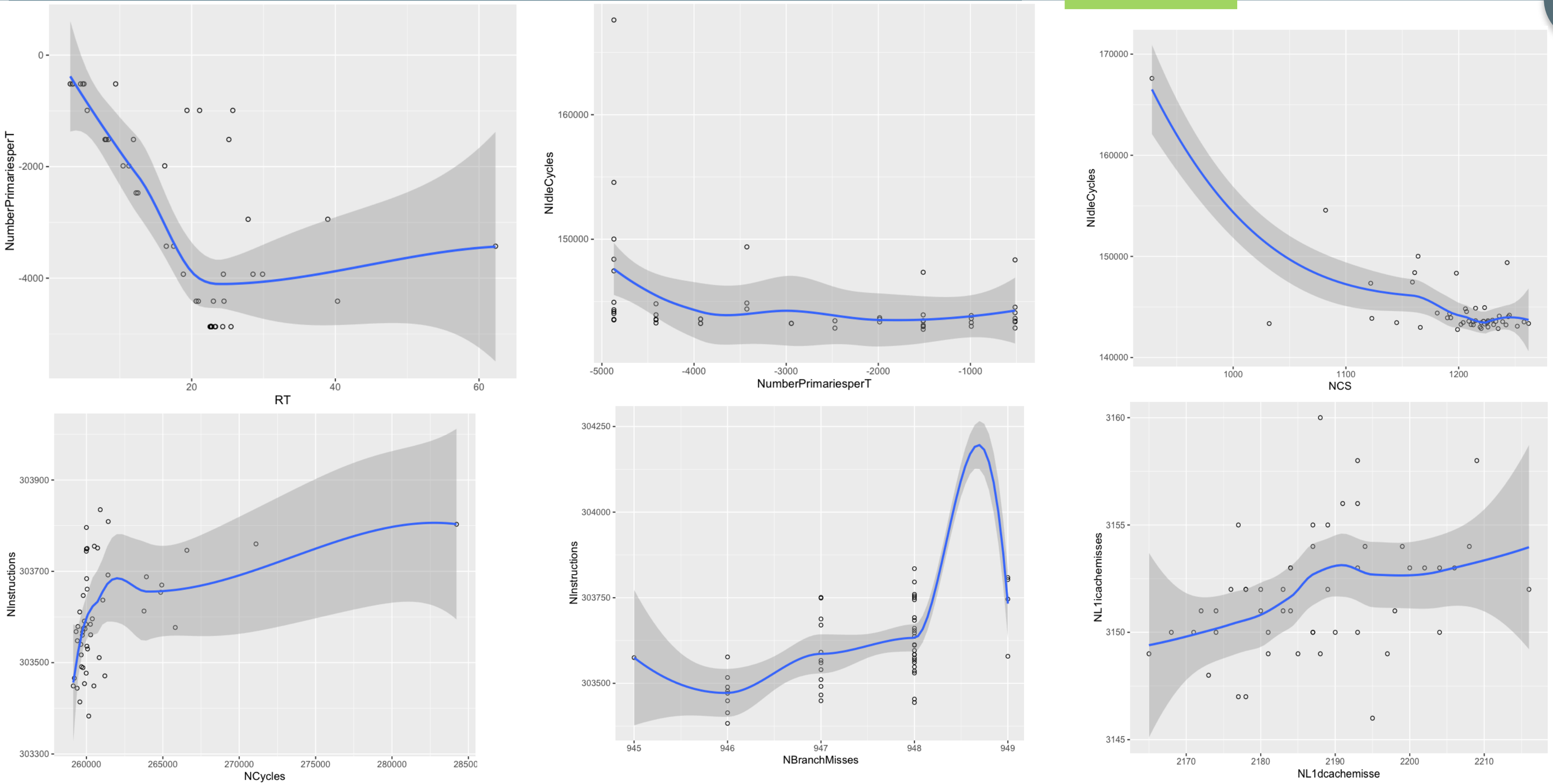
Results on first generation of GA for Example N03 (full track simulation through simplified geometry)



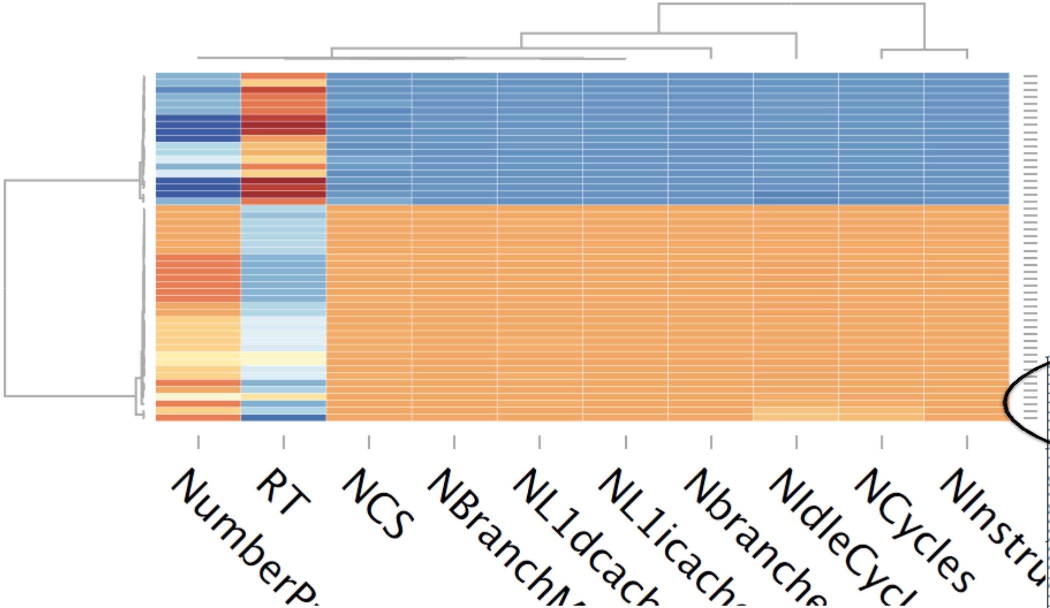


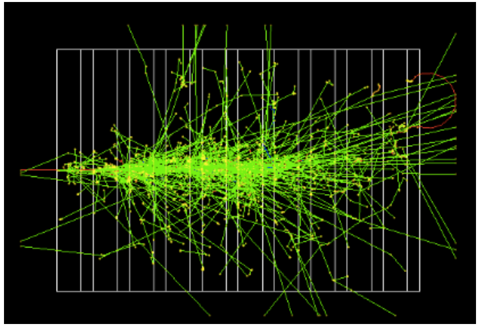


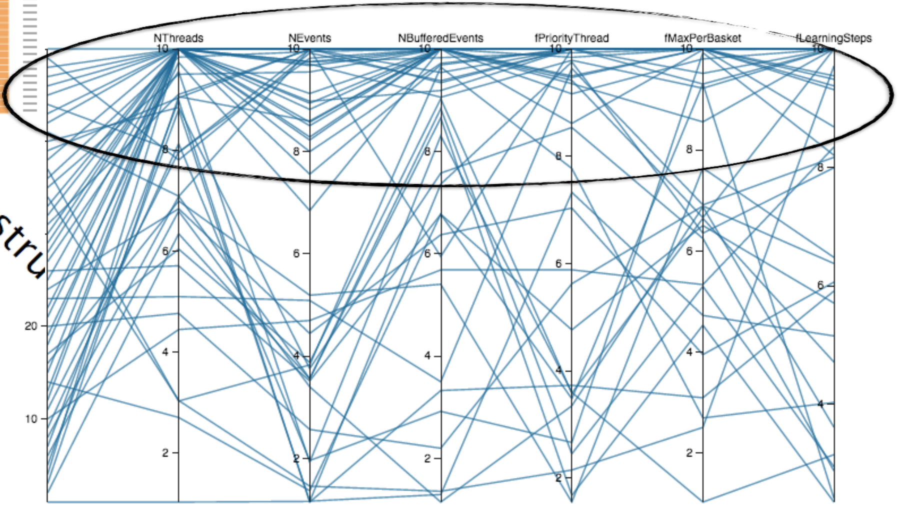
Fitness scatterplots - Core i7 – 1 Generation



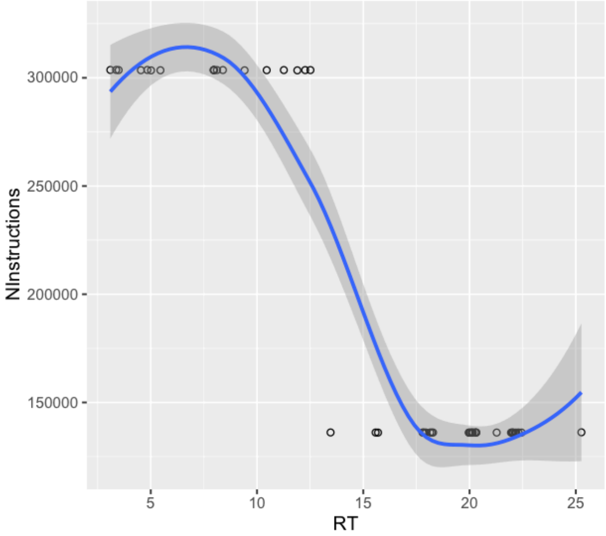
Results for Generation 13 of GA for Example N03 (full track simulation though simplified geometry)

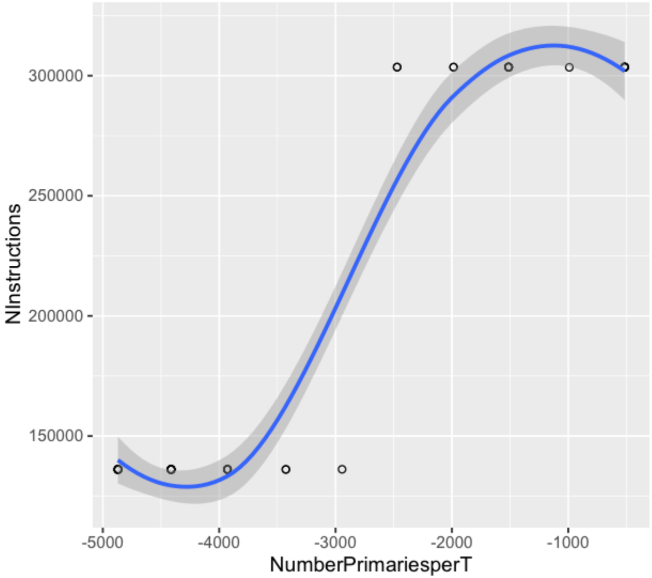


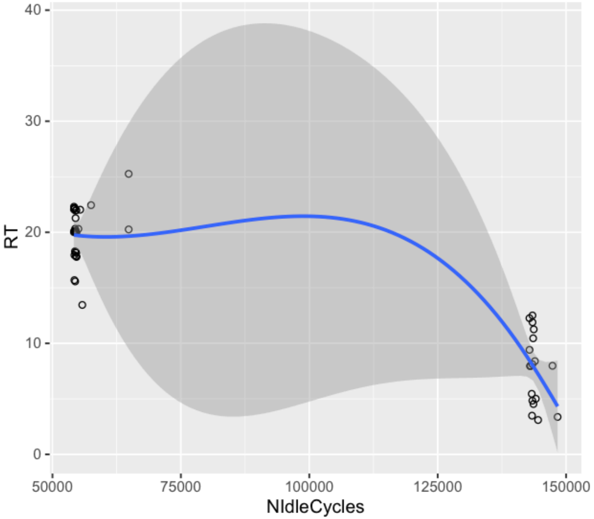


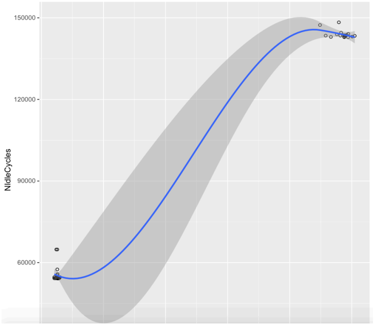


Fitness scatterplots (Pareto front approximation) – Core i7 – 13 generations



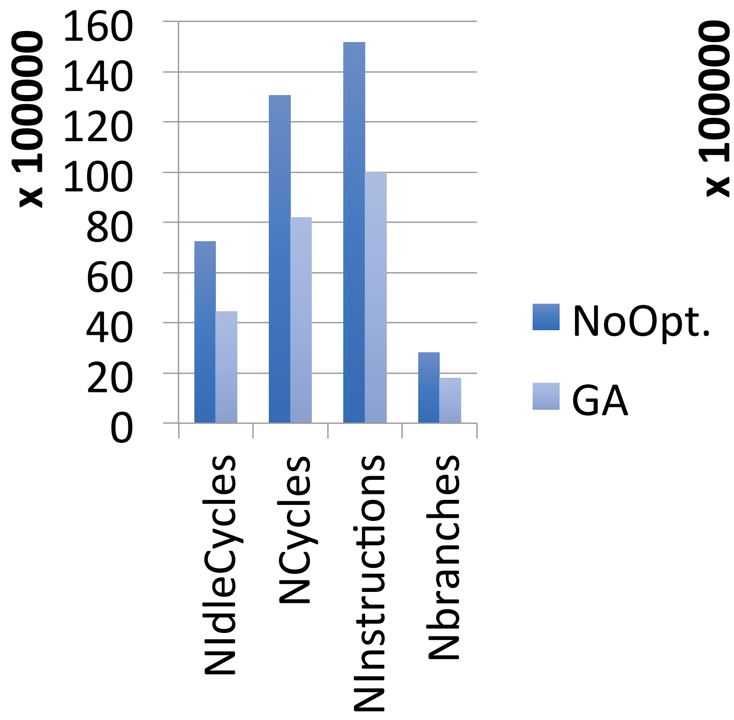


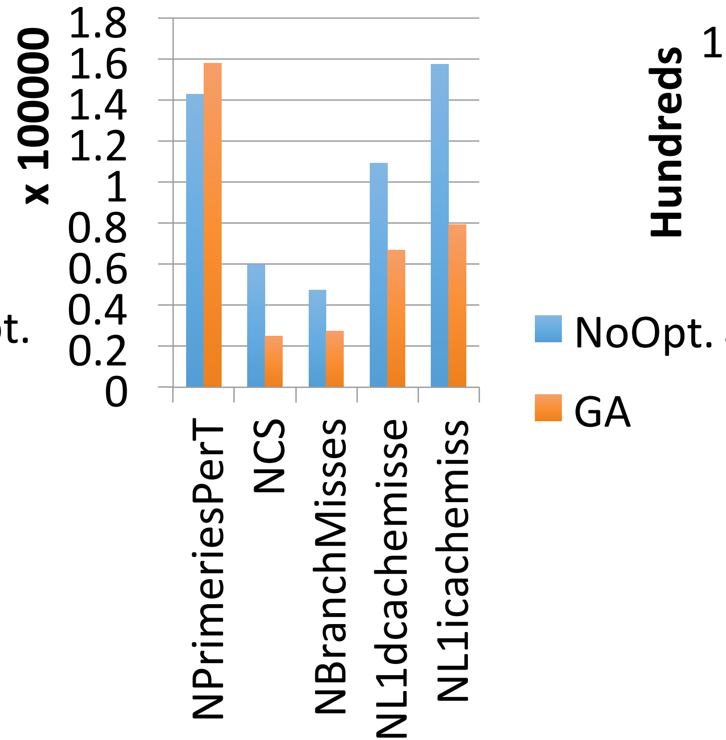


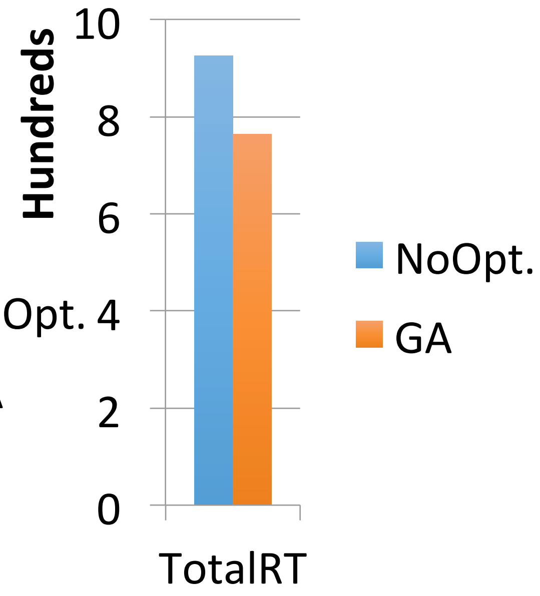


Benefits

Total runtime of batch of jobs decreased ~20 % less





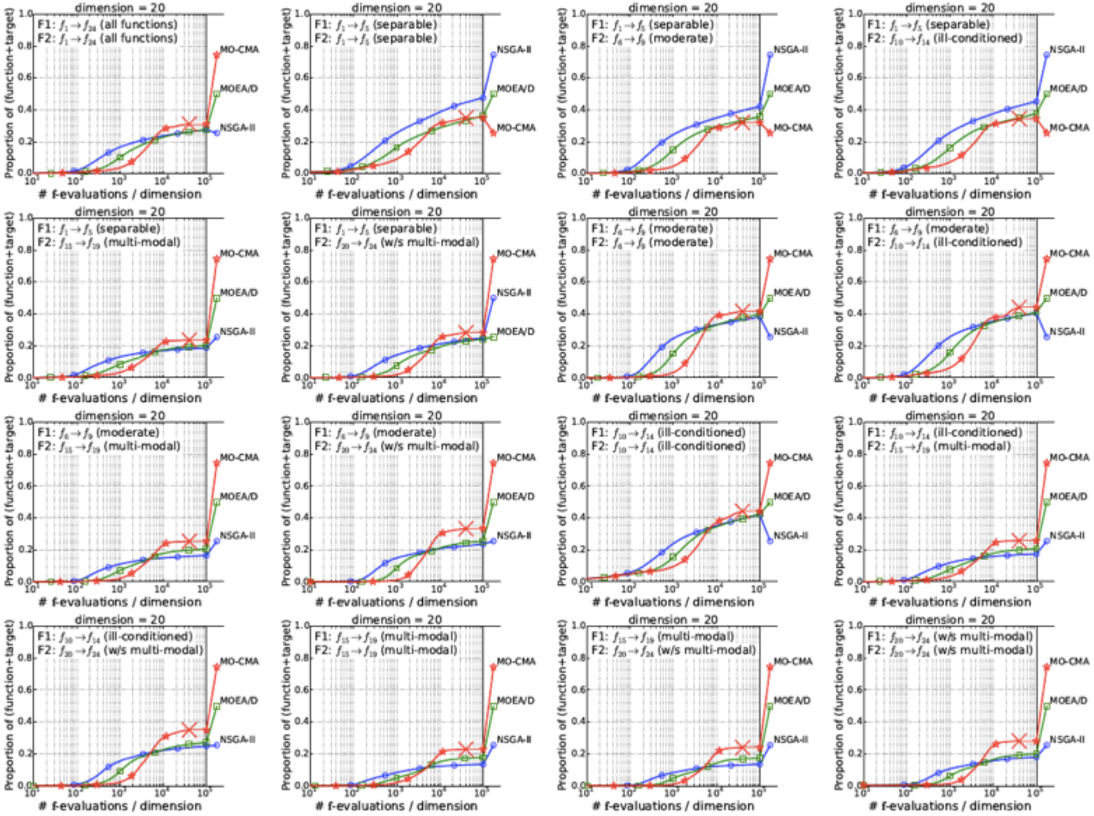


Future plans

- Work on improvement of convergence of algorithms
- Extension schema for support of heterogeneous and non-heterogeneous exascale HPC computations (MPI +X)
- Introduction of multi-layer graph spectral clustering in HPC layout of GeantV
- Testing in HPC environment with a different types of nodes: GPU, MIC, KNL and etc.

Backup

BBOB 2015 results of comparison different types of genetic algorithms



.

GeantV optimisation task

Optimization problem: we want to find extremums of fitness functions which define the performance of GeantV:

$$y^l = f^l(x_i; p_k) \quad l = 1, ..., L, \tag{1}$$

where

$x_i \ (i = 1, ..., n)$ is a vector of genes dimension n,

$p_k \ (k = 1, ..., m)$ is parameter vector, where parameters describe external factors, which can affect the performance of GeantV. For example, it could define a ”nodes” type (all m-type) for non-homogeneous cluster. In this case:

$p_k = 1$, if the k-th processor is included in cluster load,
$p_k = 0$, if the k-th processor is excluded from cluster load

Let us define uncentered data matrix of the size $M \times n$

$$\hat{X}^{(u)} = \{X^{(u)}_{\alpha,i}\} = \{x^{(\alpha)}_i\}, \tag{2}$$

where $x^{(\alpha)}_i \ (\alpha = 1, ..., M)$ is α-th sample of the population.

Then fitness functions matrix is define as matrix $M \times L$

$$F_{\alpha,l}(p_k) = \{f^l(X^{(u)}_{\alpha,i}; p_k)\}, \tag{3}$$

Uncentered PCA as operator for constrained data in GA

1). The case of the fixed value of components of the parameter vector $\vec{p}$:

To improve the convergence rate of GA we add a new operator $\hat{P}$ to a standard set of GA operator's (selection, mutation, crossing), performing uncentered PCA on the GA populations.

Convergence genetic operator

$$G_P(\vec{p}) = \underline{\hat{P}} \circ \hat{C} \circ \hat{U} \circ F(\vec{p})$$

The important objects for uncentered PCA are the matrix $W_{i,j} = (w_i)_j$ of the eigenvectors $\{\vec{w}_j\}$ and the matrix $\Lambda^{(u)}_{i,j} = \lambda^{(u)}_i \delta_{i,j}$ eigenvalues $\{\lambda^{(u)}_i\}$ of the matrix of non-central second moments,

$$\hat{T} = \frac{1}{M}\hat{X}^{(u)\,t} \cdot \hat{X}^{(u)} = \{T_{i,j}\} = \frac{1}{M}\{X^{(u)\,t}_{i,\alpha} X^{(u)}_{\alpha,j}\}$$

and

$$\hat{T} \cdot \vec{w}_j = \lambda^{(u)}_j \vec{w}_j, \quad \vec{w}^{\,t}_i \cdot \vec{w}_j = \delta_{i,j}, \quad 1 \le i,j \le n,$$

Let define the matrix $V^{(u)}_{\alpha,j} = \{(v^{(u)}_\alpha)_j\}$ of the uncentered principal components $\{\vec{v}^{(u)}_j\}$ where

$$V^{(u)}_{\alpha,j} = X^{(u)}_{\alpha,i} W_{i,j}, \quad W^t_{i,i'} W_{i',j} = \delta_{i,j},$$

For the variance of j-th uncentered principal component we obtain

$$\mathrm{Var}(\vec{v}^{(u)}_j) = \lambda^{(u)}_j - (\vec{\mu})^2 \cos^2(\vec{\mu}, \vec{w}_j)$$

Using the representation

$$V^{(u)}_{\alpha,j} = \sqrt{M}\bar{V}^{(u)}_{\alpha,i} \Lambda^{(u)\,1/2}_{i,j},$$

where

$$\Lambda^{(u)\,1/2}_{i,j} = \left[\lambda^{(u)}_i\right]^{\frac{1}{2}} \delta_{i,j}, \quad \bar{V}^{(u)\,t}_{i,\alpha} \bar{V}^{(u)}_{\alpha,j} = \delta_{i,j}.$$

we obtain the Singular Value Decomposition (SVD) of the uncentered data matrix

$$X^{(u)}_{\alpha,i} = \sqrt{M}\bar{V}^{(u)}_{\alpha,k} \Lambda^{(u)\,1/2}_{k,j} W^t_{j,i}.$$

If $\lambda^{(u)}_k \ll 1$ for $k = p+1, p+2, ..., n$ we can apply the **"eigenvalue control parameter"** approximation to get for the output data matrix $\tilde{X}^{(u)}_{\alpha,j}$ of rang p

$$\tilde{X}^{(u)}_{\alpha,i} = \sqrt{M}\left(\left[\lambda^{(u)}_1\right]^{\frac{1}{2}} \bar{V}^{(u)}_{\alpha,1} W^t_{1,i} + ... + \left[\lambda^{(u)}_p\right]^{\frac{1}{2}} \bar{V}^{(u)}_{\alpha,p} W^t_{p,i}\right).$$

The mean square error η_p for this approximation:

$$\eta_p = \frac{1}{Mn}\sum_{\alpha=1}^{M}\sum_{i=1}^{n}(X^{(u)}_{\alpha,i} - \tilde{X}^{(u)}_{\alpha,i})^2 = \frac{1}{n}\sum_{k=p+1}^{n}\lambda^{(u)}_k$$

Clustering as a part of optimized scheduling process

II). The case of the fixed value of components of the data matrix $X^{(u)}_{\alpha,i}$:

Task is define as the optimisation of parameters $\vec{p}$ for configuration of non-heterogeneous clusters with fixed α-th sample in data matrix (near Pareto front) for fitness tensor of size $L \times 2^m$

$$F^l_{p_1,p_2,\ldots,p_m} = \{f^l(x^{(\alpha)}_i; p_k)\}, \quad (1)$$

Guess: this task could be optimised using a procedure of clustering of values of fitness functions which could group in clusters of similar configuration, Pareto fronts which correspond to different values of the parameter vector $\vec{p}$.

For standard spectral clustering is used spectral theorem of Laplace operator on graphs. In our case we will use **Laplace operator on multi-layer graphs**, where each layer is numerated by values of the parameter vector $\vec{p}$, for example $(p_1 = 0, p_2 = 1, \ldots, p_m = 1)$