Implementation study of DISCOPOLIS, an algorithm for uniform sampling of metabolic flux distributions via iterative sequences of linear programs

Name: Hongxing NIU

Specialized Master in data science, Big data,

Academic year 2019-2020,
Université Libre de Bruxelles,

Promotor: Prof. Philippe Bogaerts

3BIO-BioControl, Brussels School of Engineering, Université Libre de Bruxelles

CONTENTS

ACKNOWLEDGEMENTS

It was 15 years ago that I did my last master, when I never thought I would do another one on a completely different subject after such a long time. Working on this research thesis has been a unique journey for me and I would not have been reaching my destination without a lot of people.

Frist of all, I owe a great deal of debt to my Promotor: Prof. Philippe Bogaerts, who deserves special thanks not only for his advising, but also for his unwavering support and inexhaustible supply of good ideas. His immense love and enthusiasm for science have given me more spirit and courage to accomplish this study. He is a warm friend as well, picking me up when I am down.

Moreover, I would like to thank my family for understanding me for this master study. If I could choose anybody at all to be in my corner, backing me up, through all the difficulties of life, it would be my family a hundred times over. For my son, Di, sorry for being even grumpier than normal while I wrote this thesis! And for my wife Sülen, thanks for all your support, without which I would have never started doing this master. Now, I will be clearing up all the mess on my table as I promised!

Hongxing NIU

ABSTRACT: The DISCOPOLIS algorithm via iterative sequences of linear programs constrains the flux distribution samples of a metabolic network inside the solution polytope with all the previously estimated fluxes. The solutions are weighted to ensure sampling uniformity. The algorithm configuration was investigated by tuning its two key parameters, specifically sample size (*nSample*) and discretization (*nGrid*), improve the algorithm performance. Gelman and Rubin diagnostic was used to monitor convergence of solutions and determine the *nSample* to terminate. Besides the marginal distribution, generalized variance and total sample variance were calculated to quantify statistical dispersion of solution population with changing *nGrid*. It was found in two cases that the optimal *nGrid* was in the range of 10~20 with which a neither overdispersed nor underdispersed flux distribution was achieved. At last, the mean solution representative of the core metabolic network for *E. coli.* was calculated with an optimal setting of the algorithm.

Keywords: Metabolic network, Metabolic Flux Analysis, uniform sampling, underdetermined systems, constraint-based modeling, Monte Carlo, convergence monitoring, statistical dispersion

1. INTRODUCTION

Large biochemical networks, metabolic networks in particular, can be represented by means of various graphs. In detail, a network can be modeled as a bipartite digraph which nodes are the biochemical entities and reactions and edges are the substrate or product link between an entity and a reaction. Metabolic Network Analysis is to study the properties of these graphs. Central is the use of the so-called stoichiometric matrix. Several methodologies have been established so far. The method of Metabolic Flux Analysis (MFA) (Stephanopoulos *et al*., 1998) has been widely used to determine cellular metabolic flux values based on algebraic linear equations for the mass balances of intracellular metabolites and inequality constraints for lower and upper bounds of the fluxes. In most cases, the unknown fluxes outnumber the equations, hence giving an underdetermined system of linear equations subject to constraints. (1) The standard approach is Flux Balance Analysis (Orth *et al*., 2010a), which adds an objective function (often called the Biomass function) to maximize after reduces the network system to a set of linear equations with constraints. How to choose the right objective function is still in question and the maximization of objective function can often be reached with different flux distributions, hence still resulting in an under-determined system. (2) Another approach to solve such underdetermined systems is to apply linear programming, obtaining minimum and maximum values of each flux, i.e., returning the boundaries for the fluxes through each reaction, so called Flux Variability Analysis (Mahadevan and Schilling, 2003). (3). Recently, Mhallem Gziri and Bogaerts (2019) proposed a method of Most Accurate Fluxes (MAF), which provides a unique solution of flux distribution without assumptions regarding an optimal biological behavior.

In addition to the above approaches to an underdetermined system, uniform sampling of the solution space can build marginal distributions for each flux and determine mean metabolic behavior under specific conditions and accordingly provide an unbiased characterization of the metabolic capabilities of a biochemical network. After the flux space is reduced by equality constraints, the feasible solution is contained in a convex polytope defined by the intersection of a set of half-planes corresponding to inequality constraints. Algorithms that sample the feasible region of an underdetermined linear problem in a uniform way, have already been described in the literature. The most intuitive one is the rejection technique (Rubinstein, 1982). But usually it is not well suited to complex networks as the fraction of samples to be rejected increases dramatically with the dimension of the solution space. Another approach, Hit-and-Run sampler, by Smith and Kaufman (1984, 1998) follows a random walk that can globally reach across the solution space. However, it is a well-known drawback that the Hit-and-Run sampler gets often stuck in some parts of the polytope if the solution polytope is of an irregular shape with some highly elongated directions. Recently, an improved version of the hit-and-run method, the coordinated hit-and-run with rounding method (CHRR) (Haraldsdóttir *et al*., 2017), has been added to the COBRA toolbox.

In previous work, a new approach for uniform sampling of metabolic flux distributions via iterative sequences of linear programs (DISCOPOLIS) has been proposed (Bogaerts and Rooman, 2019). It determines each sample with the following procedure: firstly choosing randomly the order of fluxes, and secondly fixing iteratively their values using uniform random sampling on intervals defined by updated linear programs. One important feature of this new approach is that it recovers uniformity by attaching weights to the samples and correcting the reduction of the iteratively updated interval of each flux during sampling. In this thesis, I further investigate some properties of the DISCOPOLIS algorithm, such as convergence and discretized sampling (the number of grid points used in the algorithm), aiming at understanding the algorithm configuration and optimizing the parameter settings.

2. THE POLYTOPE OF SOLUTIONS FOR THE FLUX DISTRIBUTION OF A METABOLIC NETWORK

2.1 Metabolic flux analysis (MFA)

We consider a metabolic network with m internal metabolites (concentrations in mol·cell^{-1}) through n metabolic reactions (fluxes in mol·cell^{-1}). The mass balances of the internal metabolites are described by the following equation

$$\dot{c} = Sv - \mu c, \quad c, v \in \mathcal{R}^{m}, S \in \mathcal{R}^{m\times n} \tag{1}$$

where S is the stoichiometric matrix and μ is the specific cell growth rate (in h^{-1}).

The following assumptions can be usually made, (i) the internal metabolites do not accumulate in cells (quasi-steady state $\dot{c} = 0$); (ii) the dilution term $-\mu c$ may be neglected compared with the reaction term Sv. As a result, Eq.(1) reduces to the system of m algebraic equations with n unknowns

$$Sv = 0 \tag{2}$$

Secondly, other linear equality constraints are used, taking into account external measurements of some exchange fluxes. As a result, all the equality constraints (including Eq.(2)) are lumped into

$$A_e v = b_e, \quad A_e \in \mathcal{R}^{n_e \times n},\ b_e \in \mathcal{R}^{n_e} \tag{3}$$

Thirdly, the metabolic fluxes are physically subject to boundary values which can be described by inequality constraints,

$$Av \leq b, \quad A \in \mathcal{R}^{n_i \times n}, b \in \mathcal{R}^{n_i} \tag{4}$$

These constraints mainly include lower and upper bounds for the fluxes but may also contain additional inequalities based on some biological assumptions, e.g. regarding overflow metabolism (Richelle *et al*., 2016; Bogaerts *et al*., 2017).

2.2 Flux variability analysis (FVA)

The system {Eqs.(2), (3)} is usually underdetermined with the unknown fluxes (*n*) outnumber the equations (n_e). One approach to solve such an underdetermined system is Flux variability analysis (FVA). The minimum and maximum range of each reaction flux is determined using a double Linear Programming problem (*i.e.* a maximization and a subsequent minimization) for each reaction of interest

$$v_{Min,Max}(i) = Min, Max[v(i)] \qquad \forall\, i \in [1, n] \tag{5}$$

subject to {Eqs.(2), (3), (4)}.

2.3 *Elimination of the equality constraints and definition of the polytope of Solutions*

Instead of considering the *n*-dimensional space of fluxes *v* subject to the underdetermined system {Eqs.(3), (4)} of n_e equalities and n_i inequalities, a reduced n_q-dimensional space of fluxes v' (with n_q = n - n_e), subject to n_i inequalities, can be defined. Indeed, let $A \in \mathcal{R}^{n_i \times n}$ be the matrix whose columns define the orthonormal basis for the null space of A_e, *i.e.* the set of all $v \in \mathcal{R}^n$ such that $A_e v = 0$. Given that $A_e A_0 = 0$, any flux distribution *v* of (3) can be decomposed into

$$v = v_0 + A_e q, \qquad A_e v_0 = b_e\,,\ q \in \mathcal{R}^{n_q} \tag{6}$$

where v_0 is a particular solution of Eq.(3). In this new space of reduced fluxes *q*, the inequalities in Eq.(4) become

$$A'q \le b' \tag{7}$$

with

$$A' = AA_0 \tag{8}$$

$$b' = b - Av_0 \tag{9}$$

A particular solution is the parsimonious solution which can be obtained by solving the quadratic program (QP)

$$v_0 = \mathrm{Min}_v \, v^T v \qquad (10)$$

subject to {Eqs.(3), (4)}.

In this new subspace of dimension n_q, the original problem is now fully described by the set of inequalities Eq.(7). The intersection of the corresponding half-planes defines the convex polytope of solutions for the reduced flux distribution q.

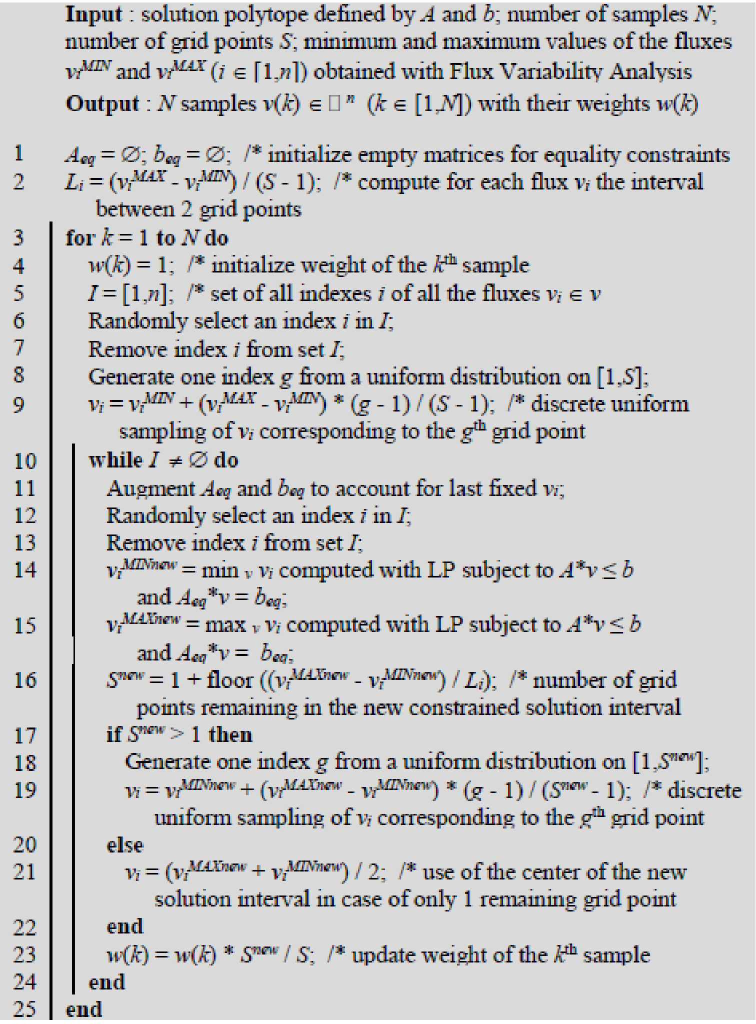
3. ALGORITHM AND METHODOLOGY

3.1 Brief review of the DISCOPOLIS algorithm

In this section, I have a brief review of the DISCOPOLIS (DIscrete Sampling of COnvex Polytopes via Linear program Iterative Sequences) algorithm which has been detailed in (Bogaerts and Rooman, 2019).

Firstly, the algorithm is setup with input of stoichiometric matrix *A* (actually, *A* is the "A''" in Eq.(8)), boundary values b (actually, b is the "b'" in Eq.(9)), parameters (sampling number *N* and number of grid points *S*), and minimum and maximum values of v_i with Flux Variability Analysis (Eq.(10)). After randomly selecting one index *i* (line 6) from vector of flux, the corresponding flux value is randomly assigned using uniform discrete sampling method (line 8) in the range of the min(v_i) and max(v_i) (line 9).

It is an iterative procedure (lines 10-24) that after fixing v_i the rest other fluxes are determined. In this procedure, the interval of v_i is renewed ([v_i^{MINnew}, v_i^{MAXnew}]) via iterative linear programs (lines 14 and 15) by Eq.(7) with the previously determined values of all the other fluxes in last iteration. Similarly, the number of grid points S^{new} in the renew interval is updated (line 16). The procedure iterates until the updated S^{new}=1 and the center of the new interval is chosen to be the value of v_i (line 21). The update in line 23 recovers uniformity by attaching weights to samples and correcting the reduction of renewed interval.



```
     Input : solution polytope defined by $A$ and $b$; number of samples $N$;
     number of grid points $S$; minimum and maximum values of the fluxes
     $v_i^{MIN}$ and $v_i^{MAX}$ ($i \in [1,n]$) obtained with Flux Variability Analysis
     Output : $N$ samples $v(k) \in \square^n$ ($k \in [1,N]$) with their weights $w(k)$

1    $A_{eq} = \varnothing$; $b_{eq} = \varnothing$;  /* initialize empty matrices for equality constraints
2    $L_i = (v_i^{MAX} - v_i^{MIN}) / (S - 1)$;  /* compute for each flux $v_i$ the interval
        between 2 grid points
3    for $k = 1$ to $N$ do
4      $w(k) = 1$;  /* initialize weight of the $k^{th}$ sample
5      $I = [1,n]$;  /* set of all indexes $i$ of all the fluxes $v_i \in v$
6      Randomly select an index $i$ in $I$;
7      Remove index $i$ from set $I$;
8      Generate one index $g$ from a uniform distribution on $[1,S]$;
9      $v_i = v_i^{MIN} + (v_i^{MAX} - v_i^{MIN}) * (g - 1) / (S - 1)$;  /* discrete uniform
          sampling of $v_i$ corresponding to the $g^{th}$ grid point
10     while $I \neq \varnothing$ do
11       Augment $A_{eq}$ and $b_{eq}$ to account for last fixed $v_i$;
12       Randomly select an index $i$ in $I$;
13       Remove index $i$ from set $I$;
14       $v_i^{MINnew} = \min_v v_i$ computed with LP subject to $A*v \leq b$
            and $A_{eq}*v = b_{eq}$;
15       $v_i^{MAXnew} = \max_v v_i$ computed with LP subject to $A*v \leq b$
            and $A_{eq}*v = b_{eq}$;
16       $S^{new} = 1 + \text{floor}((v_i^{MAXnew} - v_i^{MINnew}) / L_i)$;  /* number of grid
            points remaining in the new constrained solution interval
17       if $S^{new} > 1$ then
18         Generate one index $g$ from a uniform distribution on $[1,S^{new}]$;
19         $v_i = v_i^{MINnew} + (v_i^{MAXnew} - v_i^{MINnew}) * (g - 1) / (S^{new} - 1)$;  /* discrete
              uniform sampling of $v_i$ corresponding to the $g^{th}$ grid point
20       else
21         $v_i = (v_i^{MAXnew} + v_i^{MINnew}) / 2$;  /* use of the center of the new
              solution interval in case of only 1 remaining grid point
22       end
23       $w(k) = w(k) * S^{new} / S$;  /* update weight of the $k^{th}$ sample
24     end
25   end
```

Fig.1 Pseudo code of the DISCOPOLIS algorithm, image from (Bogaerts and Rooman, 2019)

As a result, the mean value of a uniform flux distribution is

$$\bar{v} = \sum_{k=1}^{N} p[v(k)]\, v(k) \tag{11}$$

where

$$p[v(k)] = \frac{w(k)}{\sum_{k=1}^{N} w(k)} \tag{12}$$

and

$$w(k) = \prod_{i=1}^{n} \frac{1}{v_i{}^{MAX} - v_i{}^{MIN}} \qquad \forall k \tag{13}$$

3.2 Implementation of the DISCOPOLIS algorithm

3.2.1 Ensuring good mixing and monitoring convergence of averages

The mean of sample $\bar{\boldsymbol{v}}$, is also a random variable and consequently has its own distribution. Although we skip the theoretical details and proof, we argue that under fairly general conditions, the $\bar{\boldsymbol{v}}$ by the DISCOPOLIS algorithm is converging eventually with increase of sample size because the produced chains are ergodic.

When we run the DISCOPOLIS algorithm, one basic question arises: has the chain run long enough to traverse all portions of the region of support of target distribution? In other words, how many samples (*nSample*) should be taken before to stop the DISCOPOLIS algorithm?

One obvious way to monitor convergence to the target distribution is to run multiple sequences of the chain and plot $\bar{\boldsymbol{v}}$ versus the iteration number. If they do not converge to approximately the same value, then the estimates and inferences we get from it are suspect. A commonly used approach is that of Gelman and Rubin method (Gelman and Rubin, 1992), which is based on the idea that a between-chain variance will be considerably larger than the within-chain variance if convergence has not taken place.

3.2.1.1 Univariate Gelman and Rubin diagnostic

We start off with *M* parallel chains (i.e., random seeds) of equal-length *N* over dispersed points over the support of the $\bar{v}_i$ (the *i*th element of flux vector $\bar{\boldsymbol{v}} \in \mathcal{R}^n$). The between-chain variance *B* and the within-chain *W* are calculated for each scalar summary $X = \bar{v}_i$. We denote the *q*th scalar summary in the *p*th chain by

$$X_{pq}; \qquad p = 1, \ldots, M, \qquad q = 1, \ldots, N \tag{14}$$

Thus, the subscript q represents the position in the chain, and p denotes which chain it was calculated from.

The between-chain variance B is given as

$$B = \frac{N}{M-1} \sum_{p=1}^{M} (\bar{X}_{p.} - \bar{X}_{..})^2, \tag{15}$$

where

$$\bar{X}_{p.} = \frac{1}{N} \sum_{q=1}^{N} X_{pq} \tag{16}$$

and

$$\bar{X}_{..} = \frac{1}{M} \sum_{p=1}^{M} \bar{X}_{p.} \tag{17}$$

Eq.(16) is the mean of the N values of the scalar summary in the pth chain, and Eq.(17) is the average across chains.

The within-chain variance W is determined by

$$W = \frac{1}{M} \sum_{p=1}^{M} {S_p}^2 \tag{18}$$

with

$${S_p}^2 = \frac{1}{N-1} \sum_{q=1}^{N} (X_{pq} - \bar{X}_{p.})^2 \tag{19}$$

Note that Eq.(19) is the sample variance of the scalar summary for the pth chain, and Eq.(18) is the average variance for the M chains.

Finally, W and B are combined to get an overall estimate of the variance of X in the target distribution:

$$\widehat{var(X)} = \frac{N-1}{N} W + \frac{1}{N} B \tag{20}$$

Eq.(20) is a conservative estimate of the variance of X, if the starting points are overdispersed (Gelman, 1996). In other words, it tends to over-estimate the variance.

Alternatively, the within-chain variance given by W is an underestimate of the variance of X. This should make sense considering the fact that finite sequences have not had a chance to travel all of the target distribution resulting in less variability for X. As N gets large, both $\widehat{var(X)}$ and W approach the true variance of X, one from above and one from below.

The Gelman-Rubin approach diagnoses convergence in univariate case by calculating

$$\hat{R} = \sqrt{\frac{\widehat{var(X)}}{W}}, \tag{21}$$

This is the ratio between the upper bound on the standard deviation of X and the lower bound. $\hat{R}$ in Eq.(21) is called the estimated potential scale reduction factor (*PSRF*). If the potential scale reduction is high, then the analyst is advised to run the chains for more iterations. Gelman (1996) recommends that chains would be run until for all scalar summaries are less than 1.2.

3.2.1.2 Multivariate extension

Most underdetermined problems (flux distributions) are inherently multivariate in that the goal is to sample from a multidimensional target distribution. Brooks and Gelman (1998) proposed the following multivariate extension of the univariate GR diagnostic (Vats and Knudson, 2020).

Let $\boldsymbol{F}$ be a n-dimensional target distribution with mean $\overline{\boldsymbol{v}} \in \mathcal{R}^n$ and let $\boldsymbol{\Sigma}$ be the $n \times n$ covariance matrix of the target distribution. Let $\boldsymbol{X}_{\mathrm{p1}}$,…, $\boldsymbol{X}_{\mathrm{pN}}$ be the p^{th} parallel chain; the q^{th} vector $\boldsymbol{X}_{pq} = (X_{pq1}, \dots, X_{pqn})^T$.

Similar to the univariate GP diagnostic, the mean vector of the p^{th} chain is

$$\overline{\boldsymbol{X}}_{p.} = \frac{1}{N}\sum_{q=1}^{N} \boldsymbol{X}_{pq}, \tag{22}$$

and the overall mean is

$$\overline{\boldsymbol{X}_{..}} = \frac{1}{M}\sum_{p=1}^{M} \overline{\boldsymbol{X}}_{p.}, \tag{23}$$

The within-chain variance-covariance $\boldsymbol{W}$ is determined by the sample mean of $\boldsymbol{S}_1, \dots, \boldsymbol{S}_M$, *i.e.*,

$$\boldsymbol{W} = \frac{1}{M}\sum_{p=1}^{M} \boldsymbol{S}_p{}^2, \tag{24}$$

with

$$\boldsymbol{S}_p{}^2 = \frac{1}{N-1}\sum_{q=1}^{N}(\boldsymbol{X}_{pq} - \overline{\boldsymbol{X}}_{p.})(\boldsymbol{X}_{pq} - \overline{\boldsymbol{X}}_{p.})^T, \tag{25}$$

The between-chain variance-covariance $\boldsymbol{B}$ is estimated from M chains

$$\boldsymbol{B} = \frac{N}{M-1}\sum_{p=1}^{M}(\overline{\boldsymbol{X}}_{p.} - \overline{\boldsymbol{X}_{..}})(\overline{\boldsymbol{X}}_{p.} - \overline{\boldsymbol{X}_{..}})^T, \tag{26}$$

Finally, the target covariance $\boldsymbol{\Sigma}$ is decomposed and estimated by

$$\hat{\Sigma} := \widehat{var}(\boldsymbol{X}) = \frac{N-1}{N}\boldsymbol{W} + \frac{1}{N}\boldsymbol{B} \tag{27}$$

As in the univariate case, the goal is to compare the ratio of these estimators of $\boldsymbol{\Sigma}$. However, because $\boldsymbol{\Sigma}$ is a $n \times n$ matrix, a univariate quantification of this ratio is required. Let $\lambda_{\max}(A)$ denote the largest eigenvalue of a matrix A. The multivariate potential scale reduction factor (*MPSRF*) is estimated by

$$\hat{R}^n = \sqrt{\frac{N-1}{N} + \frac{\lambda_{max}(\boldsymbol{W}^{-1}\boldsymbol{B})}{N}} \tag{28}$$

The largest eigenvalue quantifies the variability in the direction of the largest variation, the principal eigenvector of $\boldsymbol{W}^{-\mathbf{1}}\boldsymbol{B}$. The multivariate expression here is a direct generalization of the univariate *PSRF*.

3.2.2 Choosing the appropriate number of grid points

It was pointed out in (Bogaerts and Rooman, 2019) that there is a trade-off between the increase of the number of grid points *nGrid* to reach good precision, and the decrease of their number to get a reasonable fraction of samples with relatively high weights, i.e. a higher fraction of samples with significant probability densities for computing the mean Eq.(11). Accordingly, the percentage of samples whose sum of weights accounts for nearly the total (99.9%) sum of weights is quantified by

$$\sum_{k=1}^{M} p[v(k)] = 0.999,\ p[v(1)] > p[v(2)] \dots > p[v(M)] \qquad (29)$$

$$\Rightarrow\ P(99.9\%\, weights) = \frac{M}{N}$$

In this work, the parameter *nGrid* is further investigated in regard to how to choose the optimal value. In other words, it necessitates an assessment of the tuning of the DISCOPOLIS algorithm with chosen parameters (specifically *nSample* and *nGrid*).

We have monitored above with the *MPSRF* how the mean of solutions (the first moment of population) generated by the DISCOPOLIS algorithm is converging with a sample size *N (nSample)*. In the following, it will be shown that *nGrid* is a parameter affecting statistical dispersion of the generated solutions.

Dispersion (also called variability, scatter, or spread) is the extent to which a distribution of data is stretched or squeezed. One widely used measure of statistical dispersion is the variance (the second central moment of population). As illustrated in Fig.3, there are three typical patterns of dispersion: underdispersion (uniform), random, or overdispersion (clumped). When the observed variance is higher than the variance of a theoretical model, overdispersion has occurred. Conversely, underdispersion means that there was less variation in the data than predicted.

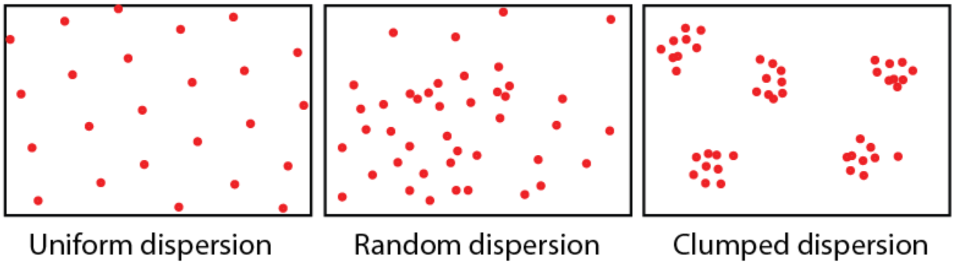


Figure 2: A schematic representation of the dispersion of individuals in a subpopulation with (from left to right) either a uniform, a random, or a clumped distribution. These are examples of absolute dispersion, or relative dispersion with respect to equal-sized quadrats. Image modified from *Population distribution* by Yerpo, CC BY-SA 4.0; the modified image is licensed under a CC BY-SA 4.0 license.

In this work, two quantities of the measure of variance are defined for multivariate data, namely "generalized variance" and "total sample variance".

3.2.2.1 Generalized variance, GV

Firstly, the weighted covariance matrix $\boldsymbol{S}$ of vector $\boldsymbol{v}$ is computed, and its elements s_{ij} are (Pozzi *et al*., 2012)

$$s_{ij} = \frac{1}{1-\sum_{k=1}^{N} p[v(k)]^2} \sum_{k=1}^{N} p[v(k)](v_i(k) - \bar{v}_i)(v_j(k) - \bar{v}_j), \quad i,j = 1, \dots, n \qquad (30)$$

Then, the *generalized variance* is obtained by the determinant of $\boldsymbol{S}$. It is a representative numerical value for the variation expressed by $\boldsymbol{S}$ and related to the multidimensional scatter of points around their mean. The *GV* has some intuitively pleasing geometrical interpretations for a given set of data (Johnson and Wichern, 2007)

$$Generalized\ sample\ variance = \det(\boldsymbol{S}) = (constant) \cdot (volume)^2 \tag{31}$$

where "volume" is the space of polytope generated by n deviation vectors $v_i - \bar{v}_i$. The volume of space occupied by the cloud of random solutions is proportional to the square root of the *GV*. The larger the *GV*, the more dispersed and the less consistent the data are.

3.2.2.2 Total sample variance, T

Another generalization of variance for multivariate data is the so called *total sample variance*. It is the sum of the diagonal elements of the sample co-variance matrix $\boldsymbol{S}$,

$$Total\ sample\ variance = s_{11} + s_{22} + \cdots s_{nn} = trace(\boldsymbol{S}) \tag{32}$$

The problem with the total variance is that it pays no attention to the correlation structure of the solutions.

4. RESULTS AND DISCUSSION

4.1 Toy Example

This toy example consists of a simple two-dimensional convex polytope (i.e., a polyhedron) for which the mean of a genuine uniform sampling can easily be computed through the rejection algorithm. It will show that the DISCOPOLIS algorithm provides accurately the mean of the genuine uniform sampling.

Let us consider the polytope defined by fluxes $\bar{v}^T = [v_1 \;\; v_2]$ belonging to the intersection of half-lines by Eq.(3) with

$$A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix},\; b = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 2 \end{bmatrix} \tag{33}$$

4.1.1 Random solutions and means

Mean of the flux distribution of Eq.(33) was accurately determined by the rejection algorithm with 5,000 uniformly distributed random samples generated. The mean of the distribution is equal to $\bar{v}^T = [0.78 \;\; 0.45]$, shown by green crosses in Fig.3.

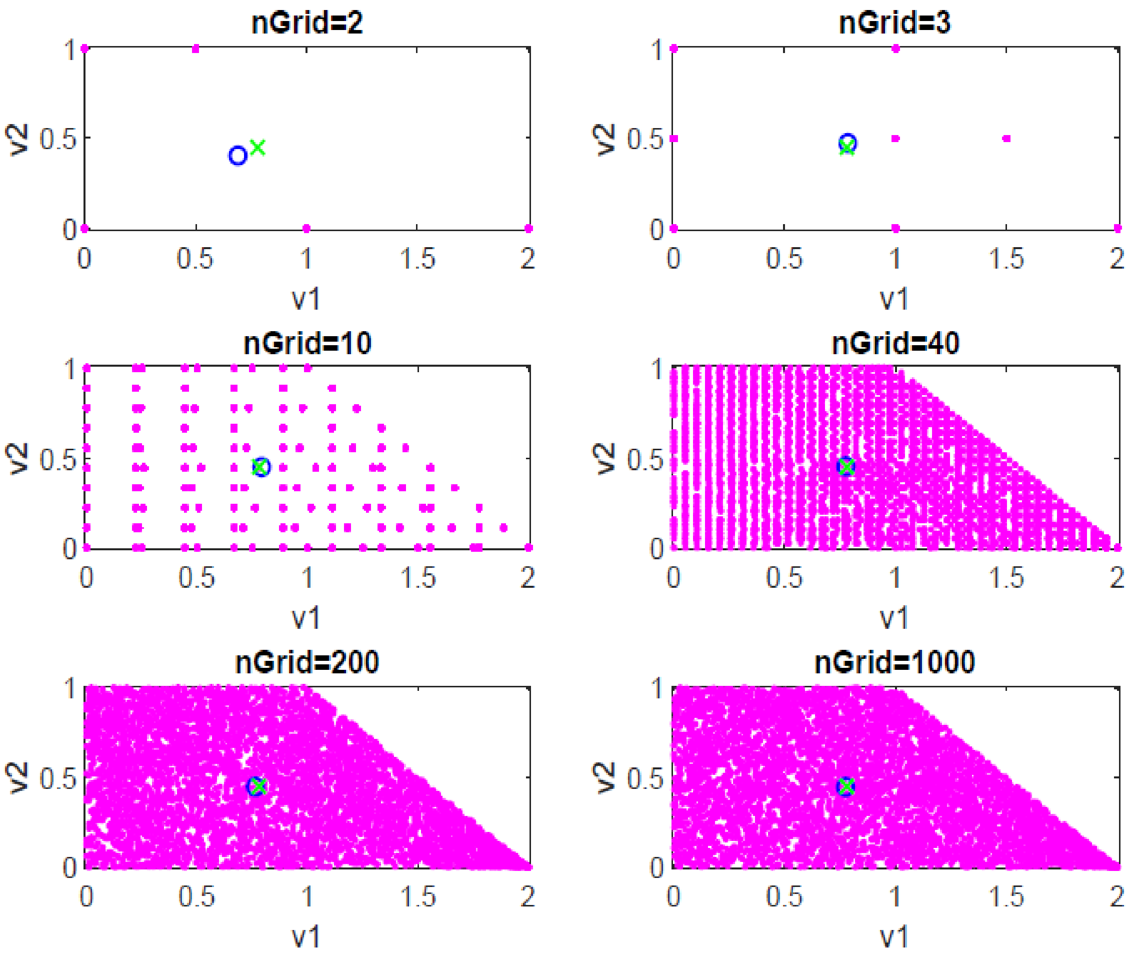


Fig.3. Solutions (magenta dots) of Eq.(33) by DISCOPOLIS with 5000 random samples taken uniformly. Six subplots corresponding to results with 2, 3, 10, 40, 200, 1000 grid points, respectively; the means under each scenario represented by blue circles. $\bar{v}^T = [0.78\ \ 0.45]$, the mean of the flux distribution by the rejection algorithm (Fig. 1), shown by green crosses.

The results provided with the DISCOPOLIS algorithm (with different numbers of grid points and seed values for the random generator) and represented in Fig.3. They show that the mean of the genuine uniform sampling performed with the rejection algorithm is recovered with very high accuracy by the DISCOPOLIS algorithm except under the scenario of 2 grid points. Moreover, the results are robust with respect to random seeds (results not shown).

4.1.2 Monitoring convergence of means

The main purpose for uniform sampling is to get random solutions in a polytope. Afterwards, we analyze and explore the characteristic of the population distribution (mean, moments, quantiles, etc.) in which we are interested.

In this work, the means of $\overline{\boldsymbol{v}}$ of the generated random solutions for {Eqs. (7), (8), (9)} are averaged over the number of samples (iteration). 10 parallel $\overline{\boldsymbol{v}}$ sequences of length N are generated by DISCOPOLIS with 10 random seeds. Distinct scenarios are produced by choosing different numbers of grid points. As an example with one random seed, sequences of the means $\overline{\boldsymbol{v}}$ are tracked over iteration, as shown in Fig.A1 in Appendix. The means $\overline{\boldsymbol{v}}$ stabilize after around one hundred iterations, indicating that the sequences are not sensitive to a random seed or to a starting state.

Furthermore, the univariate PSRF of each flux and multivariate PSRF of overall fluxes are calculated and plotted in Fig.5 to monitor if and when the means $\overline{\boldsymbol{v}}$ converge. Note, two sequences of values for PSRF are calculated for v_1 and v_2; each is used separately to monitor the convergence of corresponding flux. The MPSRF is obtained by treating the $\overline{\boldsymbol{v}}$ in form of a vector and shows how $\overline{\boldsymbol{v}}$ converge in two-dimensional space. It is found that the sequences of $\overline{\boldsymbol{v}}$ converge faster when the discretization is smaller than 40 by 40 grids. If we choose $nGrid \leq 40$, we can stop running simulations with good confidence of convergence over 5000 iterations when $(\mathrm{M})\mathrm{PSRF} < 1.2$. By comparison, it probably requires more iterations to guarantee the means of fluxes converge when each flux in solution polytope is discretized with a bigger number of grid points than 40 used in the algorithm.

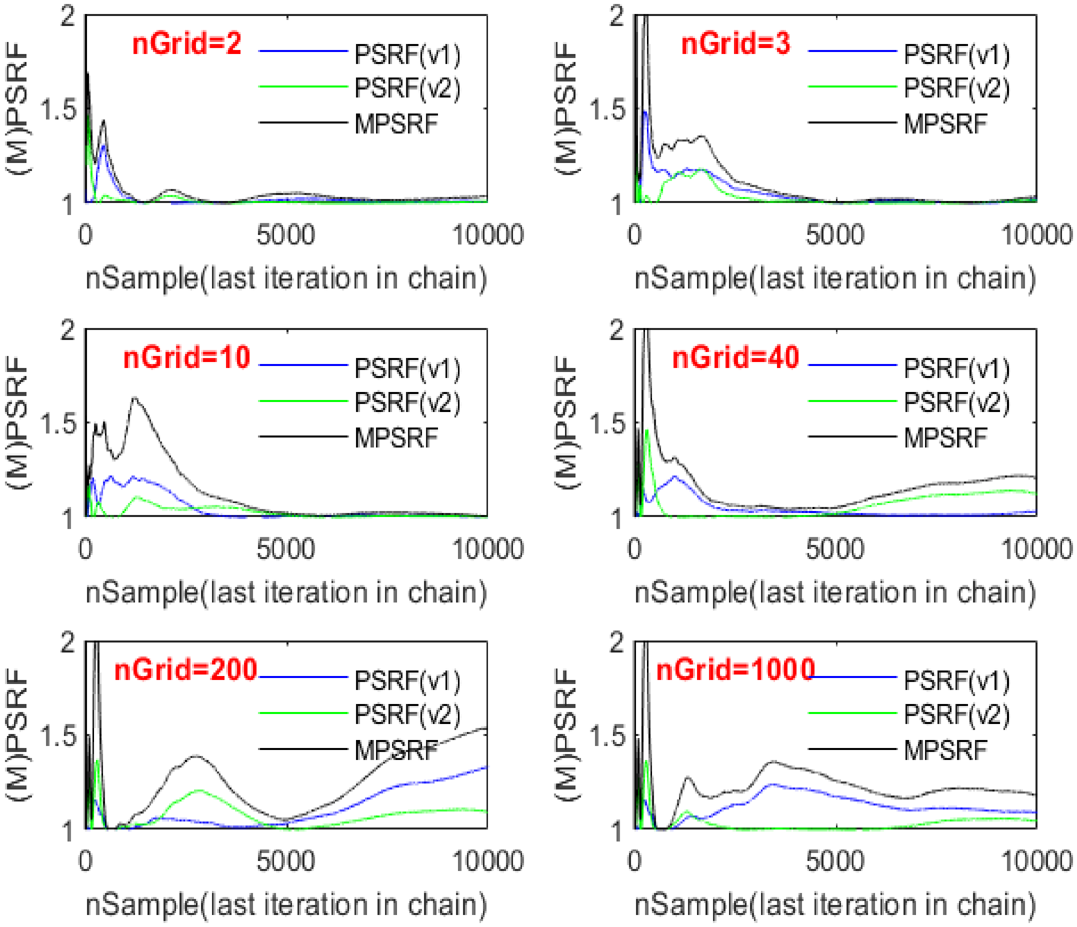


Fig.4. The sequences of values for PSRF and MPSRF indicate that they approach 1~1.1 over 5000 iterations under scenarios of *nGrid*=2, 3, or 10, showing fast convergence.

4.1.3 Choosing an appropriate number of grid points to get a "fit" target distribution of solution population

As shown in Fig.5, the "spatial" distributions of weighted values of v_1 and v_2 imply different degrees of dispersion of solution population under different scenarios. Similar phenomena are further observed through the marginal distributions of v_1 and v_2 in Fig.6. To quantify the degree of dispersion, in Fig.7 the previously defined two measures, "generalized variance" (*GV*) and "total sample variance" (*T*), are calculated and plotted against *nGrid* values with changing *nSamples*, where 10 random seeds are taken for each scenario. One interesting founding is that both *GV* and *T* drop drastically when the *nGrid* values increase from 2 to 20 and then stabilize onward, regardless of the sample size. However, the percentages of samples whose sum of weights accounts for nearly the total (*P (99.9% weight)*) do not reduce a lot from 100% in this simple toy example. Furthermore, it is observed that the more samples taken the more robust the results are. These results demonstrate that *GV* and *T* are well suited to quantify the dispersion of population generated by the DISCOPOLIS algorithm.

It is understandable that the population distribution generated with *nGrid*=2 is stretched with a largest degree of overdispersion because most of thesolutions are corner point, endpoint and vertex of polytope and are clumped (duplicated) together. In order to avoid generating a population distribution with such overdispersion, the appropriate *nGrid* values can be set in a range of 10~20.

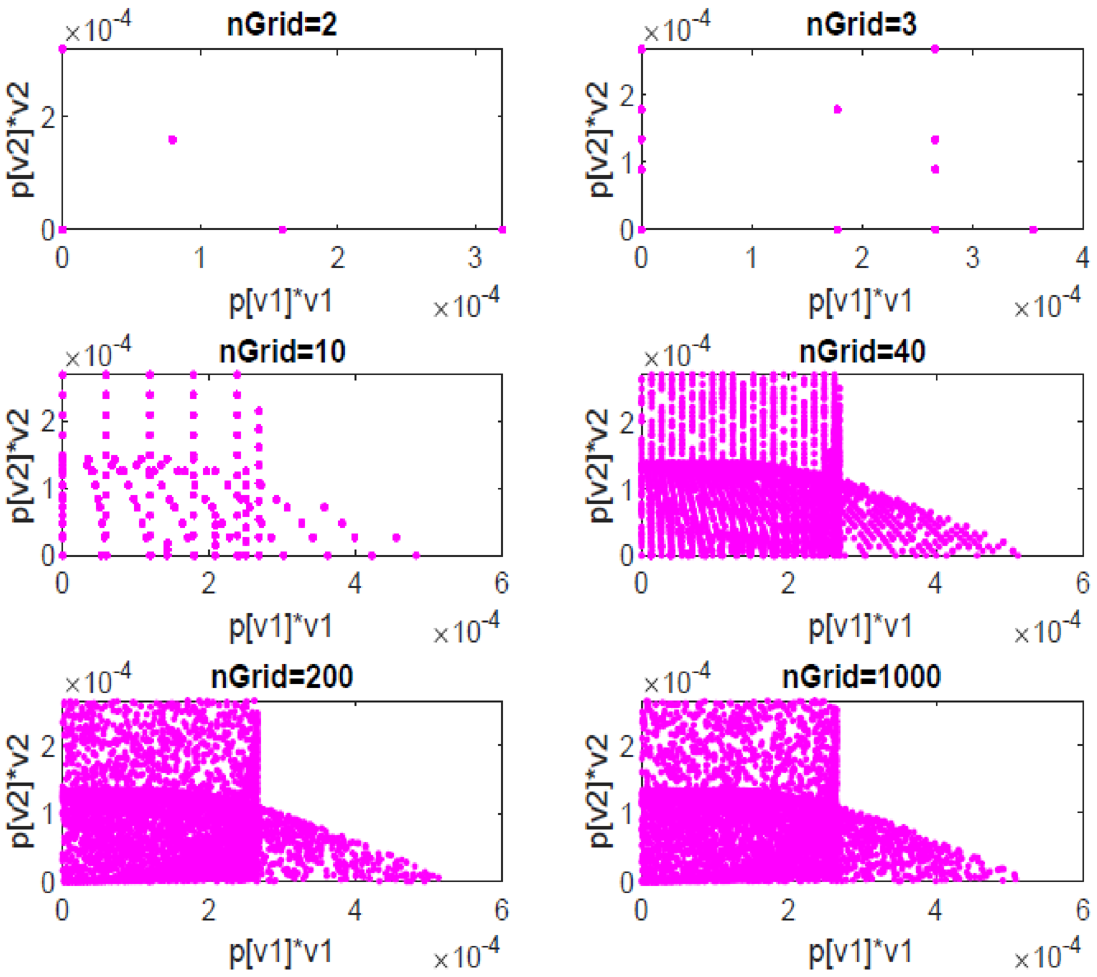


Fig.5. Distributions of weighted flux values of the toy example under different scenarios by the DISCOPOLIS with 5000 random samples taken.

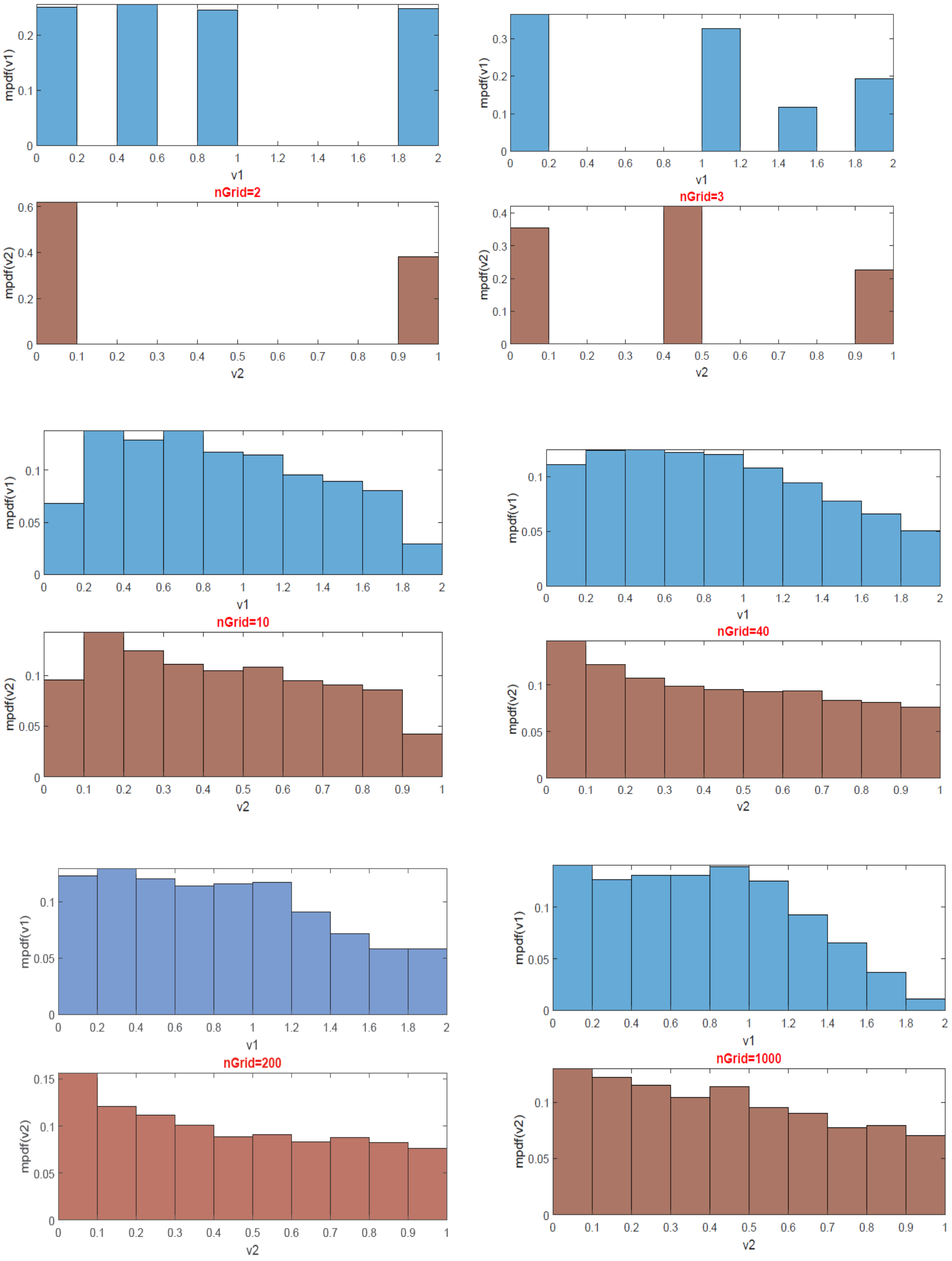
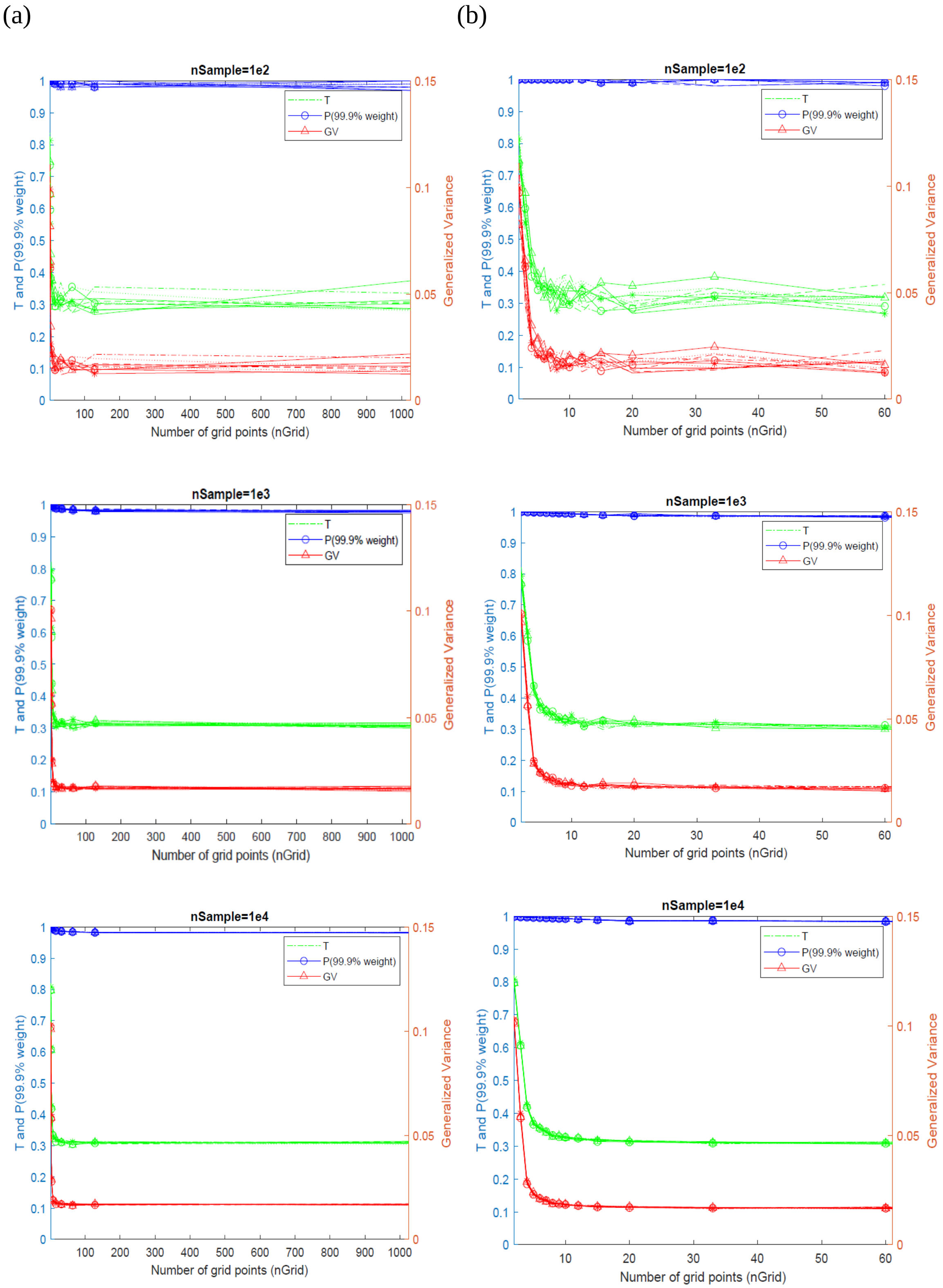


Fig.6. Marginal distributions of the fluxes of the toy example under different scenarios by the DISCOPOLIS with 5000 random samples taken.


(a)
(b)
nSample=1e2
nSample=1e3
nSample=1e4
T
P(99.9% weight)
GV
T and P(99.9% weight)
Generalized Variance
Number of grid points (nGrid)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.05
0.1
0.15
100
200
300
400
500
600
700
800
900
1000
10
20
30
40
50
60

(continued)

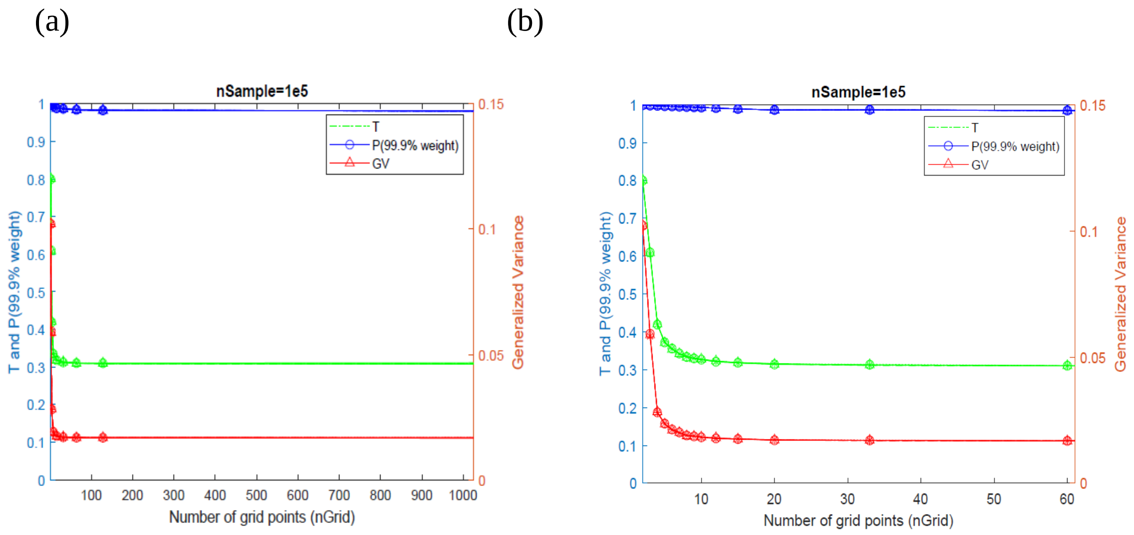


Fig.7. “Generalized variance” (GV) and “total sample variance” (T), are calculated and plotted against *nGrid* values with changing samples. The percentages of samples whose sum of weights accounts for nearly the total (P (99.9% weight)) weights are also shown. In all cases, ten random seeds are taken to under each scenario. The suplots in column (b) are the zooms for column (a).

4.2 Core Metabolic Network of *Escherichia coli*

This second case study is to analyze the core metabolic network of *E.coli* (Orth *et al*., 2010b). The COBRA model is available in the COBRA toolbox (*Ecoli_core_model.mat*). It consists of 95 fluxes with upper and lower bounds. Other details are explained in (Bogaerts and Rooman, 2019). In short, a matrix $A' \in \mathcal{R}^{172\times 23}$ was obtained through elimination of equality constraints, defining the solution polytope in Eq.(7).

4.2.1 Monitoring convergence of means

Firstly, the means of the flux network are computed against samples (iterations) with the DISCOPOLIS algorithm. Figure A2 in **Appendix** shows the means of each flux over

1000 iterations with *nGrid*=2, 10, or 100. Generally speaking, the curves of $\bar{v}_i$ traverse the solution interval (*i.e.*, $\max(v_i) - \min(v_i\,)$) after initialization and then tend to stabilize over 1000 iterations with all the degrees of discretization, indicating the sequences are in a trend towards converging. It seems that it converges the fastest with *nGrid*=2 amongst the three cases.

Secondly, in a similar way to the simple toy example, the Gelman-Rubin approach is adopted, calculating the *PSRF* and *MPSRF* to monitor the convergence of the means of flux distribution for the core metabolic network of *E.coli*. In Fig.8, we show the plots of *PSRF* and *MPSRF* for each iteration of the sequences where 10 random seeds are taken. If all the chains of $\bar{v}_i$ (i.e., means for *i*th flux corresponding to one dimension of the vector $\bar{\boldsymbol{v}}$, shown individually by blue lines in Fig.8) are reducing to 1.2 (logarithm value≈0.08), we confirm the means converge. From Fig.8, it seems it takes 5,000 iterations or even more before the $\bar{\boldsymbol{v}}$ are getting to converge. The chains seem to converge faster with *nGrid*=2 than with other degrees of discretization. Obviously, the *MPSRF* values (shown by black lines in Fig. 8) are significantly larger than any of the individual *PSRF* in this high-dimensional example with $\dim(q) = 23$, as the largest eigenvalue in Eq.(28) quantifies the variability in the direction of the largest variation, thus leading to a needlessly conservative termination criterion. It is likely the reason the multivariate *PSRF* has not found large practical use in literature. Moreover, we observe that chains of $\bar{\boldsymbol{v}}$ converge more slowly when solution interval of each flux is highly discretized (such as *nGrid*=100), which is similar to what we have found in the toy example.

In addition, other statistics are estimated (Robert and Casella, 2010), including the “effective number of samples” and “autocorrelation times” to diagnose if the chains are well mixed (results not shown).

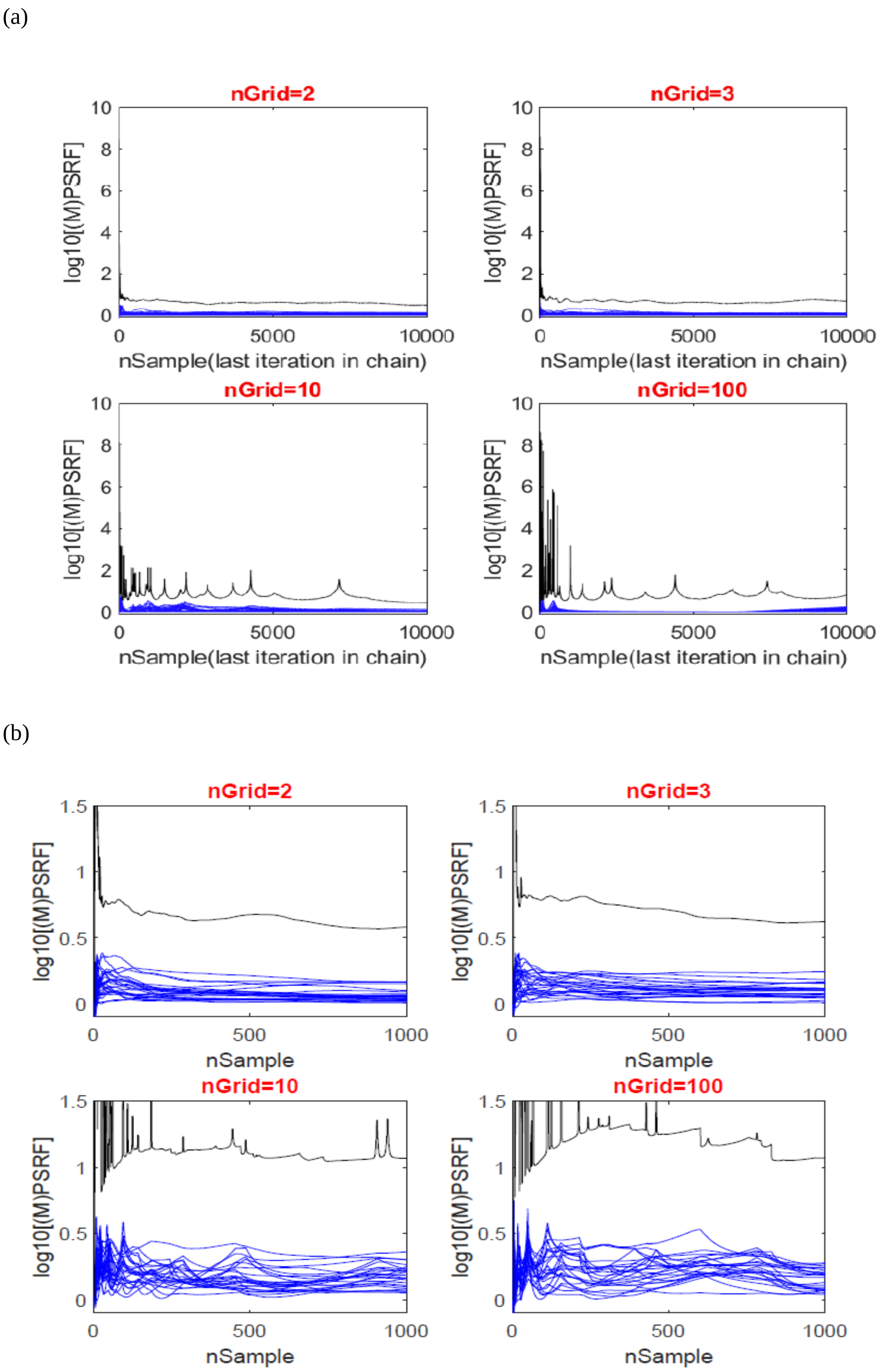


Figure 8. Convergence monitoring of means obtained by the DISCOPOLIS algorithm with changing nGrid with respect to *nSample*. (a) over 10,000 iterations; (b) the zoom for (a) over 1000 iterations.

4.2.2 Choosing an appropriate *nGrid* to get a “fit” flux distribution

The objective in this subsection is to decide an optimal discretization of solution interval in the algorithm. Here, a “fit” distribution means apparent lack of pattern or predictability in solutions, i.e., individual solution is random.

Similar to the approach used in the toy example, we firstly check the marginal distributions of each flux by simulated solutions under different scenarios. The results are shown in Figure A3 in Appendix, where 10 random seeds and 10,000 samples are taken for each scenario. There are two extreme cases: one is approximating solution interval as a binary variable (*nGrid*=2) and another is discretizing interval at a very high level of granularity (*e.g.*, *nGrid*=1000). When *nGrid*=2 (and *nGrid*=3), the solutions may move to the tails of distribution corresponding to corner points, endpoints and vertices of polytope, resulting in a heavy-tailed distribution. These results may lead to somewhat overestimating the extreme metabolic patterns (phenotypes) of organism under normal conditions. When *nGrid*=1000, the solution distribution with 10,000 random samples is more squeezed, exhibiting probably a regular pattern (solutions are peaked). Due to the computation limit of my personal computer, it is not investigated if this regular pattern resists when samples further increase to 100,000 or even more.

Furthermore, two measures *GV* and *T* are calibrated against *nGrid* to quantify the statistical dispersion of distribution generated by the DISCOPOLIS algorithm with changing discretization. Samples are chosen differently (100, 1000, 10000), producing several scenarios, and 10 random seeds are taken for each scenario. From the plots of *GV* and *T* in Fig.9, it is observed that the statistical dispersion by either *GV* or *T* shrinks rapidly when the *nGrid* values increase from 2 to 20 and then stabilizes onward till 100.

There exists high likely a *nGrid* in range of 10~20, which is an optimal option for discretization of the DISCOPOLIS algorithm. Similar to the toy example, it seems the optimal *nGrid* exists regardless of sample size as well in this *E.coli* case study. By comparison, the *P (99.9% weight)* immediately dropped below 0.1 when *nGrid*≥20. Since the obtained results are not very robust in this work, we cautiously conclude that the dispersion maybe reduce (or just randomly vary) further at very high level of discretization (*e.g*., nGrid=1000), leading to under-dispersion. As indicated by Figure A3 in **Appendix**, almost one sample may be "responsible" for the total weights when *nGrid* is very large in a complex problem.

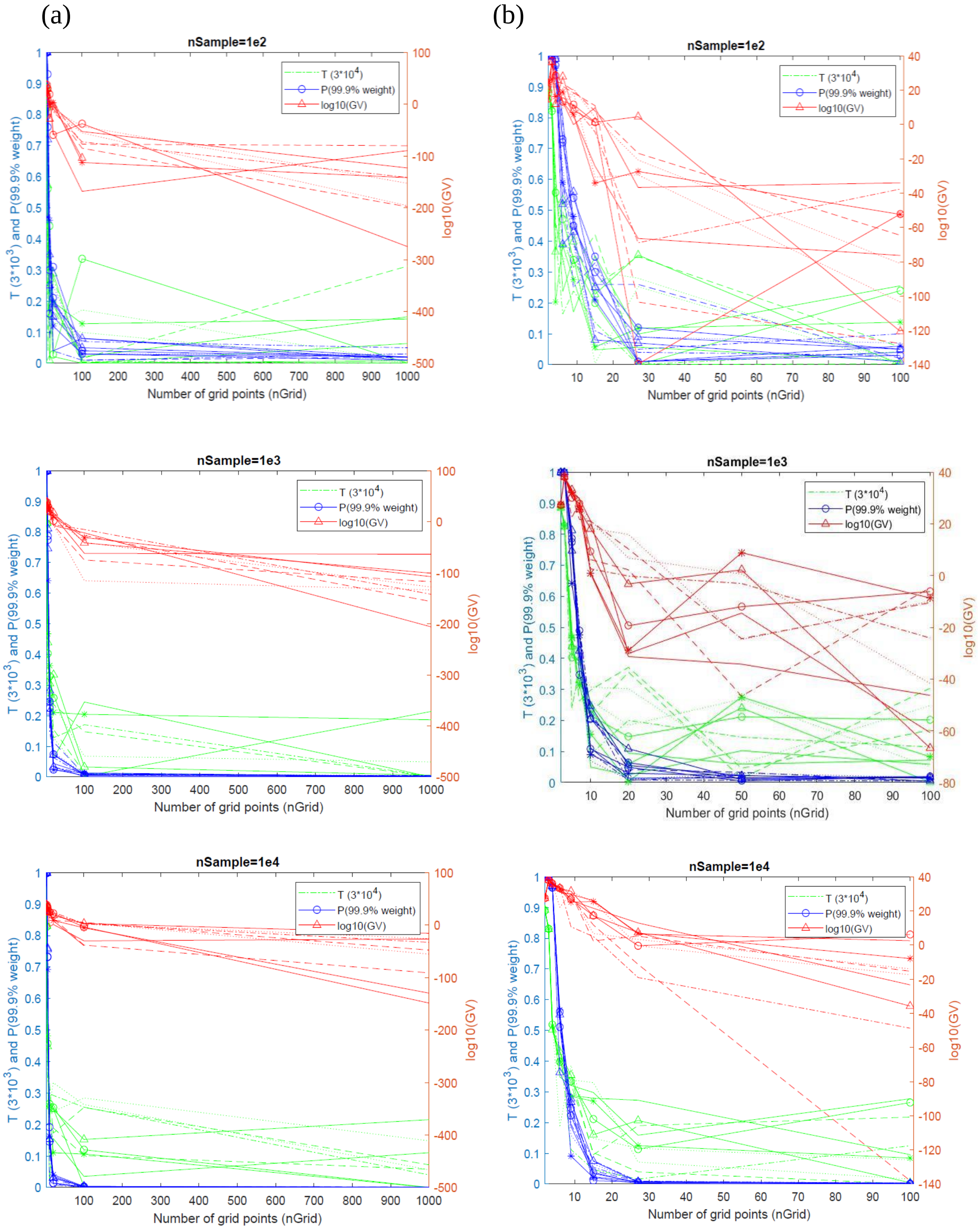


Fig.9. “Generalized variance” (GV) and “total sample variance” (T), are calculated and plotted against *nGrid* values with changing samples. The percentages of samples whose sum of weights accounts for nearly the total (P (99.9% weight)) weights are also shown. In all cases, ten random seeds are taken under each scenario. The subplots in column (b) are the zooms for column (a).

4.2.3 Plot of mean values with optimal setting of the algorithm

As investigated above, the setting of the parameters in the DISCOPOLIS algorithm has a significant impact on its performance in practice. We configure the algorithm with *nGrid*=10 in this work and use the PSRF approach (≤1.2) to decide when to stop running simulation. Blue circles in Fig.10 are corresponding to the flux means of the flux distribution within in the *E.coli* core metabolic network with the DISCOPOLIS algorithm of optimal configuration (with around 10,000 samples taken).

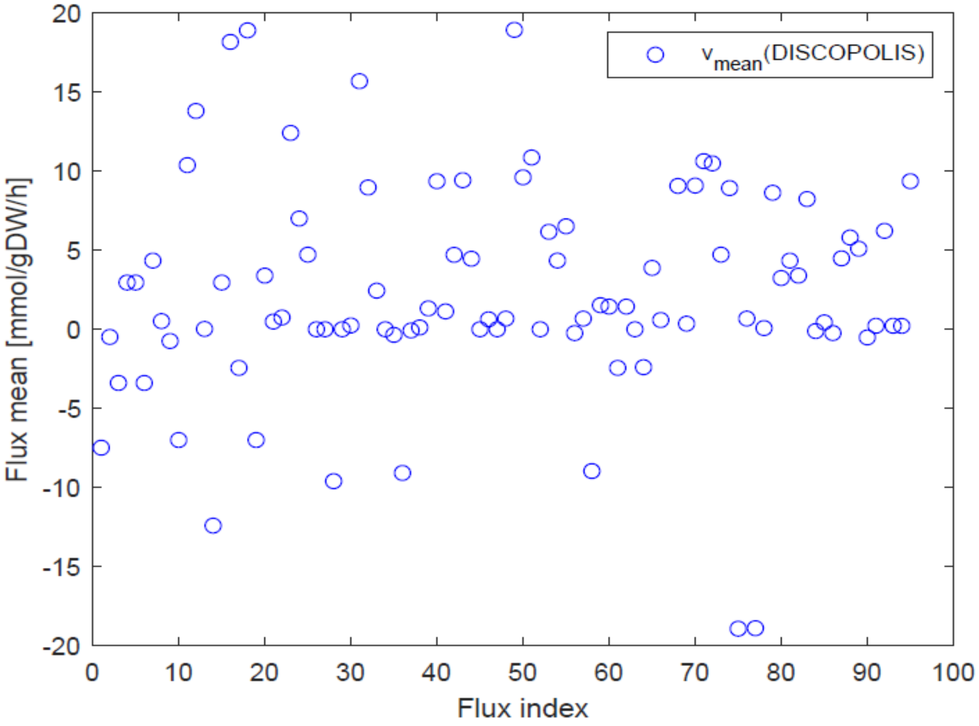


Fig.10. Flux mean values of the flux distribution within in the *E.coli* core metabolic network, obtained with optimal setting of the DISCOPOLIS algorithm, shown by blue circles

5. CONCLUSIONS AND PERSPECTIVES

The DISCOPOLIS algorithm iteratively uses linear programs for constraining the flux distribution samples of a metabolic network inside the solution polytope, taking into account all the previously estimated fluxes. The solutions are weighted to ensure sampling uniformity. The toy example showed that the mean of a genuine uniform distribution could indeed be recovered.

The configuration of DISCOPOLIS algorithm was investigated, *i.e.*, two key parameters, specifically sample size and discretization, were tuned to improve its performance. Firstly, the means of flux distribution were a focus to calculate the trend of *PSRF* against sample size to monitor if and when the results converge; secondly, two measures (*GV* and *T*) were calculated to quantify statistical dispersion of solution population generated by the DISCOPOLIS algorithm with changing number of grid points on each flux interval. It was found in the two cases the optimal *nGrid* would be in the range of 10~20 regardless of sample size with which we get a neither over-dispersed nor under-dispersed flux distribution; lastly, the mean solution representative of the core metabolic network for *E. coli.* was obtained with an optimal setting of the DISCOPOLIS algorithm.

There are still some interesting questions to address or clarify in future work,

1) Firstly, the generalized sample variance *GV* is unduly affected by the variability of measurements on a single variable. Secondly, an equal discretization along each flux should be done after it is normalized. So, the flux rescaling will have a certain impact on the results.
2) We should apply the DISCOPOLIS algorithm to more complex networks to check the conclusions obtained, especially the optimal *nGrid*. If some conclusions really

remain unchanged under different scenarios and with different networks, can they be mathematically proven?

3) The concept of a “fit’ flux distribution is still not quite well understood and defined. Besides *GV* and *T*, other measures, such as information entropy (Lubomir Kostal *et al*., 2013) and dispersion index (Cox and Lewis, 1966) will be used in future to understand the distribution from some different aspects.

4) The slow convergence of the algorithm (especially with the Gelman and Rubin diagnostic) is a notable drawback. How to “accelerate” its convergence?

Symbols and Abbreviations

nGrid (or S)	number of grid points
nSample (or N)	number of samples
E.coli	*Escherichia coli*
PSRF	univariate potential scale reduction factor
MPSRF	Multivariate potential scale reduction factor
M	parallel chains (i.e., random seeds)
B	the between-chain variance
W	The within-chain variance
$\lambda_{\max}(A)$	the largest eigenvalue of a matrix A
dim(q)	dimension of reduced flux vector
GV	generalized variance
T	total sample variance
$v(k)$	metabolic flux (sample at k iteration)
$\bar{v}_i$	mean of metabolic flux i
$w(k)$	weight of $v(k)$
$p(k)$	probability of $v(k)$
v_i^{MIN} , v_i^{MAX}	minimum, maximum value of metabolic flux i
v_i^{MINnew} , v_i^{MAXnew}	updated minimum, maximum value of metabolic flux i

References

Bogaerts Ph., Mhallem Gziri, K., and Richelle, A. (2017). “From MFA to FBA: Defining linear constraints accounting for overflow metabolism in a macroscopic FBA-based dynamical model of cell cultures in bioreactor”. J. Process Control, 60, 34-47.

Bogaerts Ph., Rooman M. (2019) , DISCOPOLIS: an algorithm for uniform sampling of metabolic flux distributions via iterative sequences of linear programs. IFAC-PapersOnLine 52 (26), 269-274.

Brooks S.P. and Gelman A. (1998). “General methods for monitoring convergence of iterative simulations”. Journal of Computational and Graphical Statistics. 7, 434-455.

Christian P. Robert and George Casella. (2010). “Introducing Monte Carlo Methods with R”. pp. 255-257. Springer.

Cox D. R.; Lewis P.A.W. (1966). “The Statistical Analysis of Series of Events”. London: Methuen.

Gelman, A. and D. B. Rubin. (1992). “Inference from iterative simulation using multiple sequences (with discussion),” Statistical Science, **7**:457–511.

Gelman, A., F., Bois, Y., and Jiang, J. (1996), “Physiological Pharmacokinetic Analysis using Population Modelling and Informative Prior Distributions,” Journal of the American Statistical Association, 91, 1400–1412.

Haraldsd óttir H., Cousins B., Thiele I., Fleming R., Vempala S. (2017). “CHRR: coordinate hit-and-run with rounding for uniform sampling of constraint-based models”. Bioinformatics. 33 (11), 1741-1743.

https://www.ecologycenter.us/population-dynamics-2/dispersion-of-individuals.html. Last Updated on Thu, 29 Jun 2017.

Johnson Richard Arnold and Wichern Dean W. (2007) Chapter 3. Applied multivariate statistical analysis. Pearson; 6th Edition.

Lubomir Kostal , Petr Lansky, Ondrej Pokora. (2013). Measures of statistical dispersion based on Shannon and Fisher information concepts. Information Sciences 235:214–223

Kaufman, D., and Smith, R. (1998). Direction choice for accelerated convergence in hit-and-run sampling. Oper. Res., 46 (1), 84-95.

Klamt S., and Stelling J. (2003). Two approaches for metabolic pathway analysis? Trends in Biotech., 21 (2), 64-69.

Mahadevan R., and Schilling C. (2003). The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. Metab. Eng., 5, 264-276.

Mhallem Gziri K., and Bogaerts Ph. (2019). Determining a unique solution to underdetermined metabolic networks via a systematic path through the Most Accurate Fluxes. Accepted in IFAC-PapersOnLine, Proceedings of the 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS 2019).

Orth J., Thiele I. and Palsson B. (2010a). What is flux balance analysis? Nature Biotechnol., 28 (3), 245-248.

Orth J., Fleming R., and Palsson B. (2010b). Reconstruction and use of microbial metabolic networks: the core Escherichia coli metabolic model as an educational guide. EcoSal Plus, 1(10).

Pozzi ., Matteo T. Di, Aste T. (2012). "Exponential smoothing weighted correlations", The European Physical Journal B, Volume 85, Issue 6, 2012. DOI: 10.1140/epjb/e2012-20697-x.

Richelle A., Mhallem Gziri K., and Bogaerts Ph. (2016). A methodology for building a macroscopic FBA-based dynamical simulator of cell cultures through flux variability analysis. Biochem. Eng. J., 114, 50-61.

Rubinstein R. (1982). Generating random vectors uniformly distributed inside and on the surface of different regions. Eur. J. Oper. Res., 10 (2), 205-209.

Schellenberger J., Que R., Fleming R., Thiele I., Orth J, Feist A., Zielinski D., Bordbar A., Lewis N., Rahmanian S., Kang J., Hyduke D., Palsson B. (2011). Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. Nat. Protoc., 6 (9), 1290-1307.

Smith, R. (1984). Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. Oper. Res., 32 (6), 1296-1308.

Stephanopoulos G., Aristidou A., and Nielsen J. (1998). Metabolic engineering: Principles and methodologies, chapter 8. Academic Press, San Diego.

Vats D., Knudson Ch. (2020). “Revisiting the Gelman-Rubin Diagnostic” .arXiv:1812.09384v2 [stat.CO] 13 Apr 2020.

Appendix

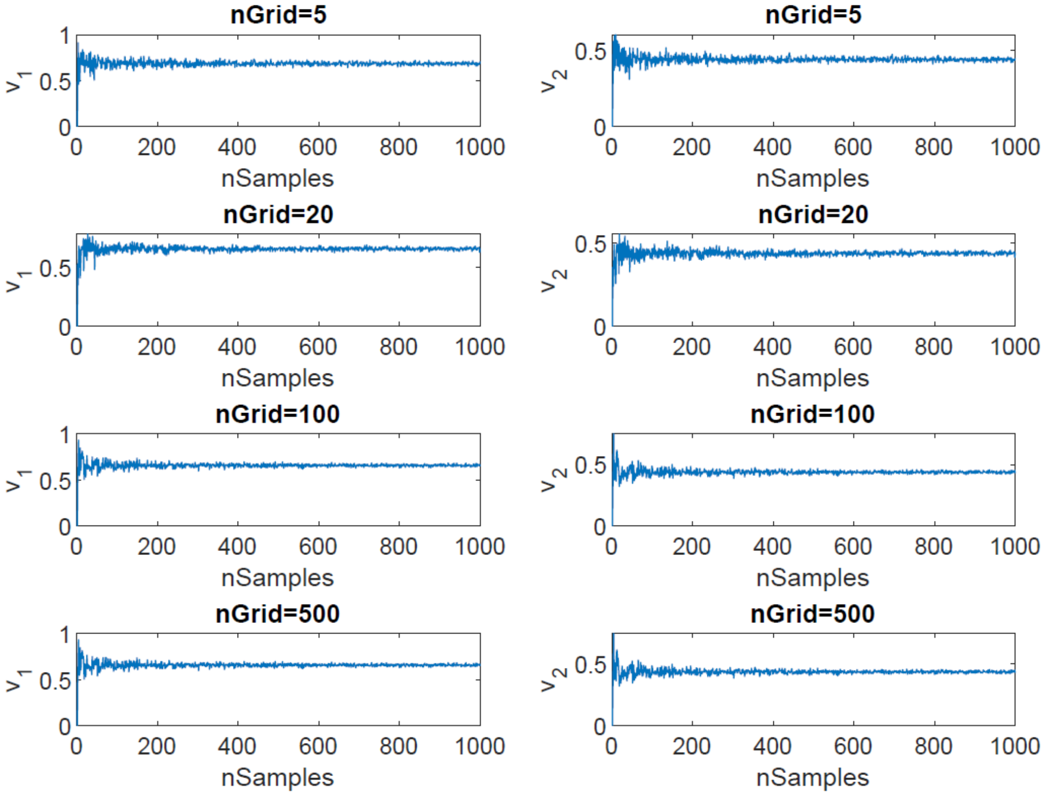
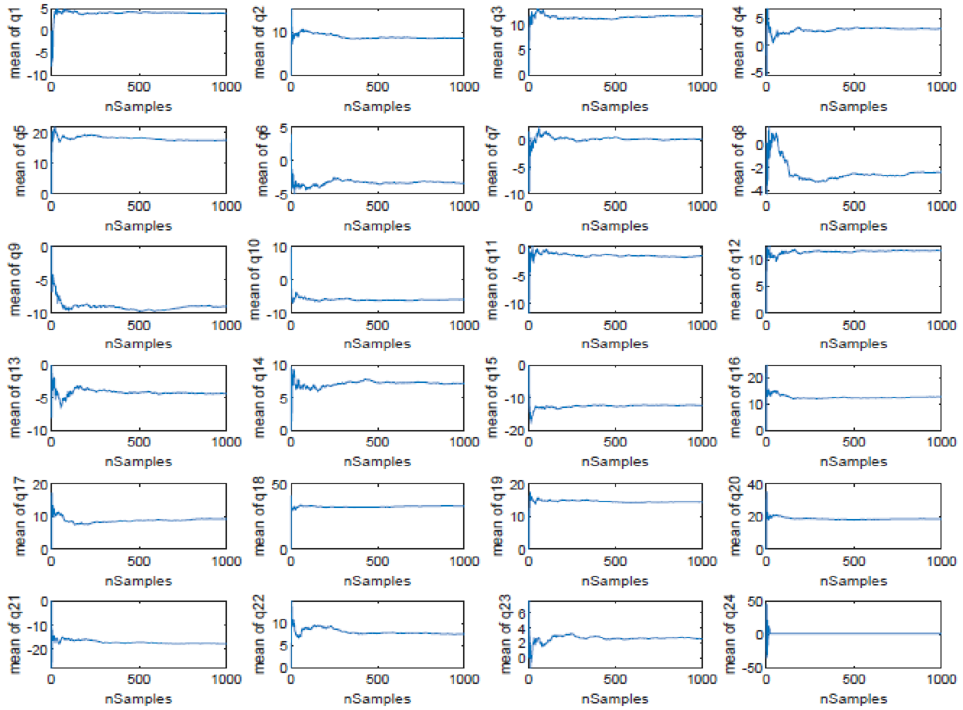
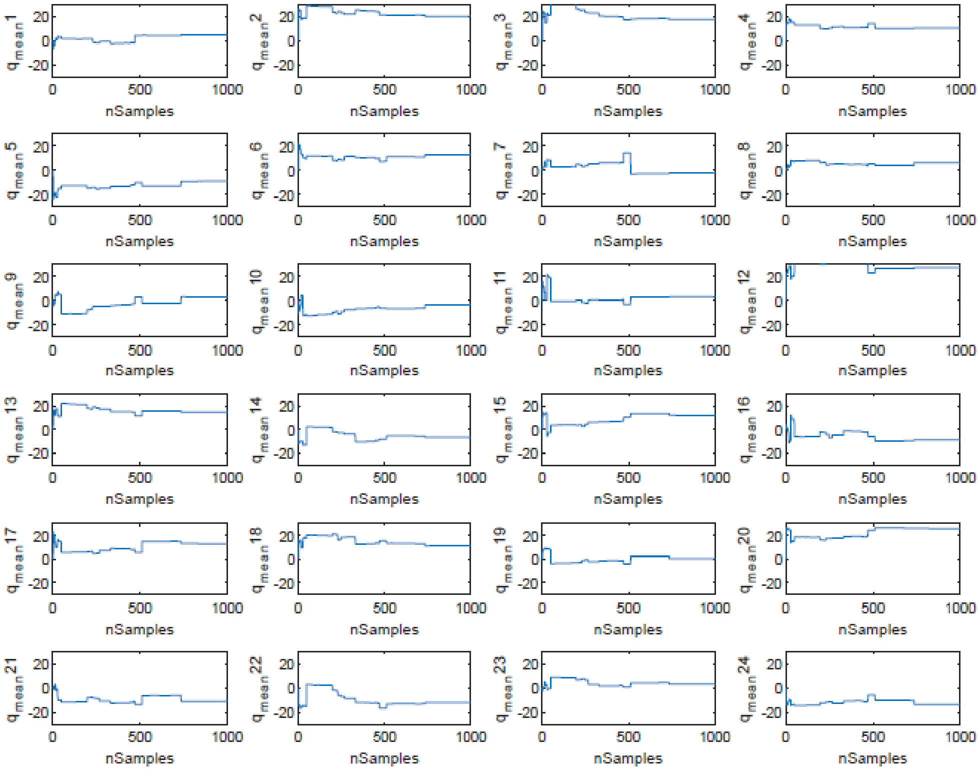


Figure A1. This shows the v1 and v2 means of the toy example over sample size, *nSamples*. Each subplot corresponds to one scenario generated by DISCOPOLIS with one distinct number of grid points, *nGrid*.

(a) nGrid=2



(b) nGrid=10



(continued)

(c) nGrid=100

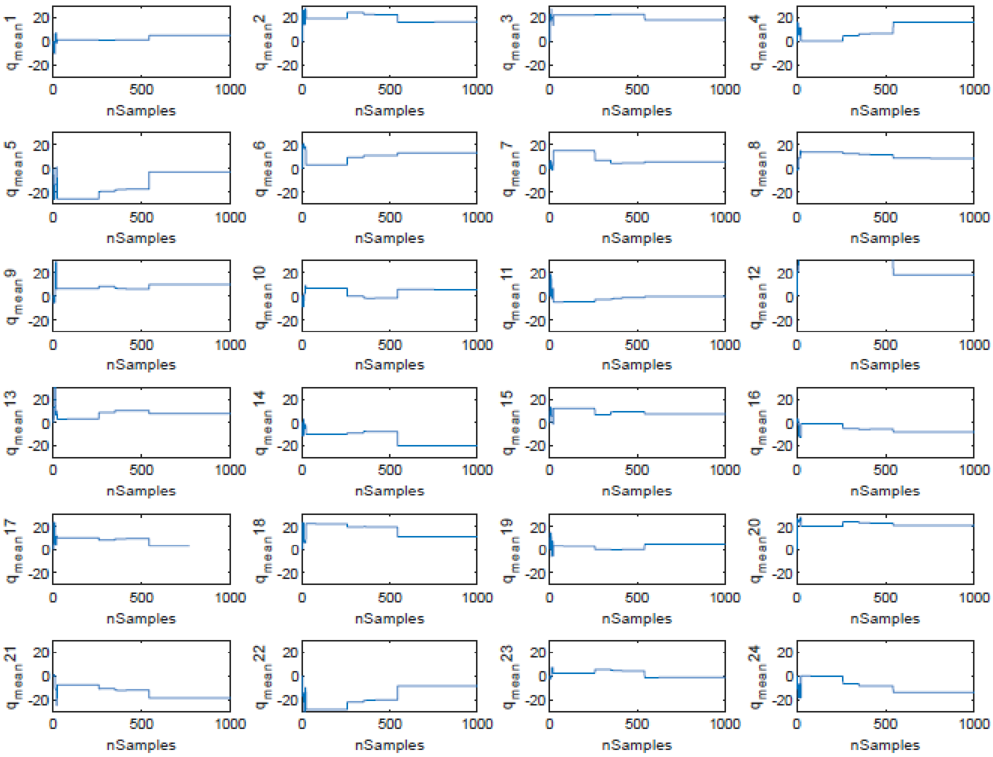
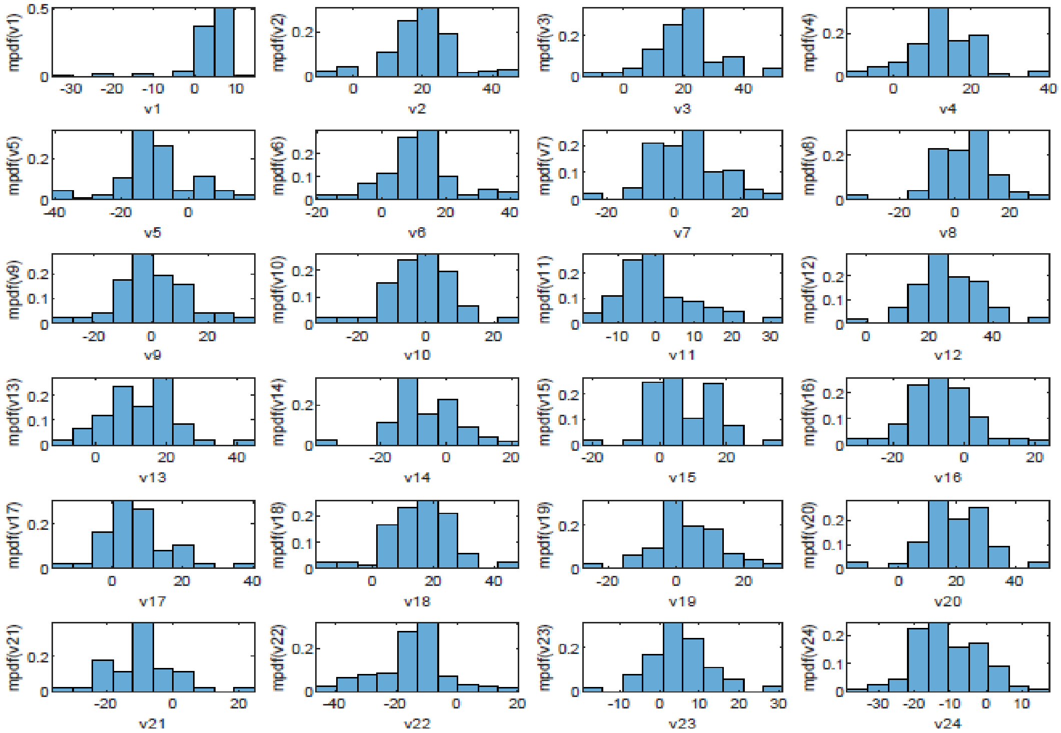
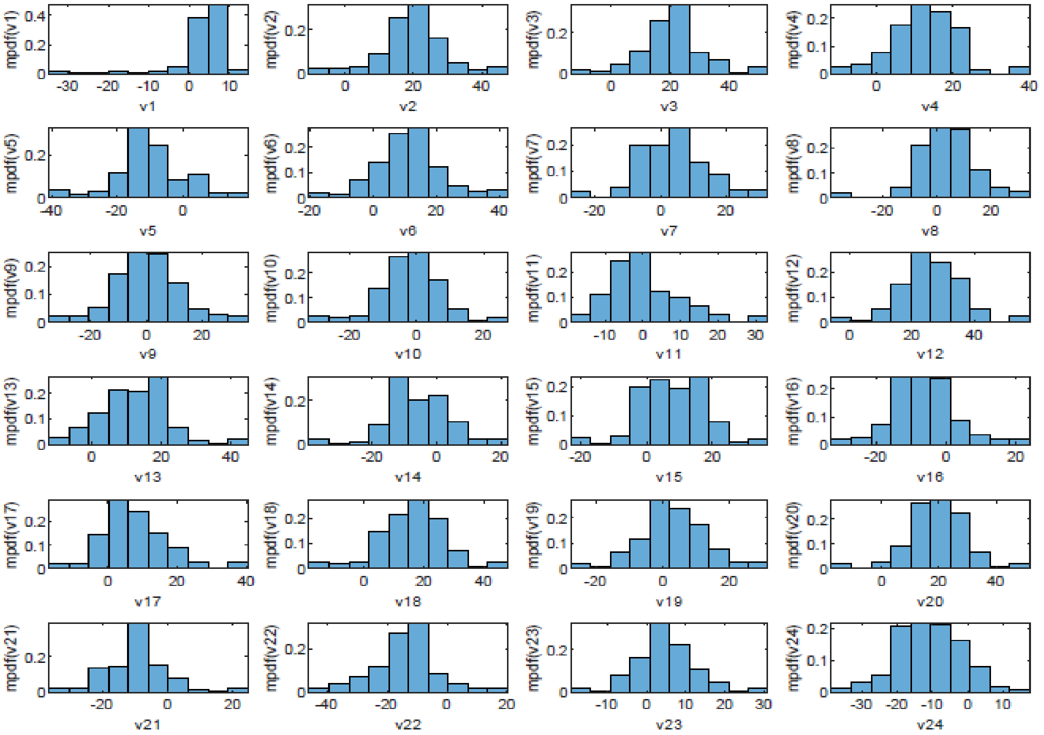


Figure A2. This shows the flux means of core metabolic network of *Escherichia coli* over sample size *nSample*, generated by the DISCOPOLIS algorithm with *nGrid*=2, 10,100 for subplot (a), (b), (c), respectively.

(a) nGrid=2, the solutions could move to the tails

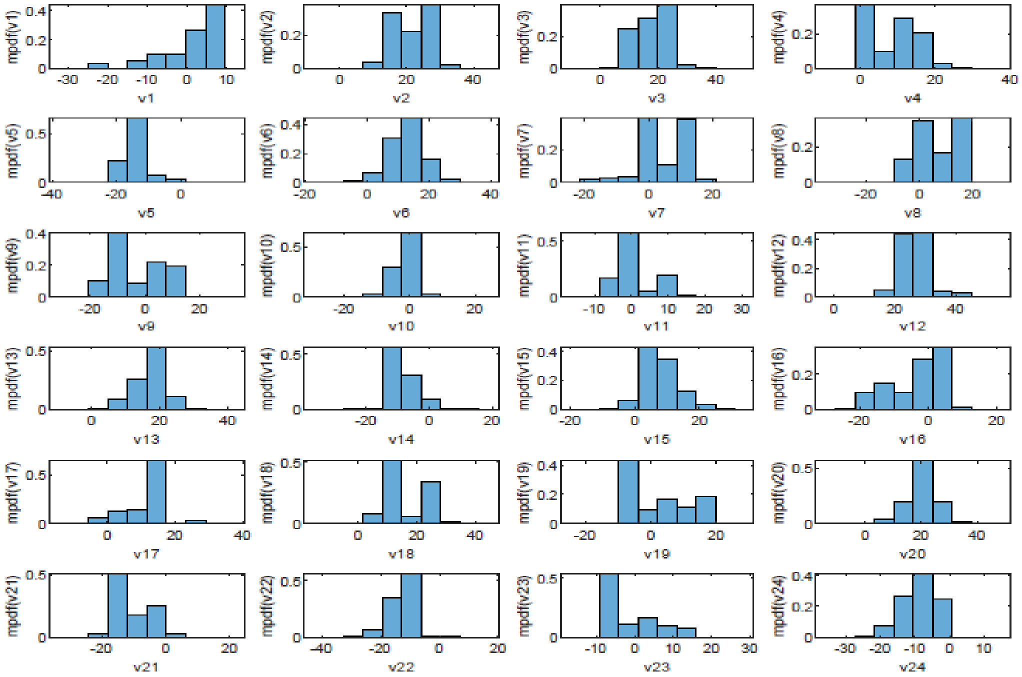


(b) nGrid=3

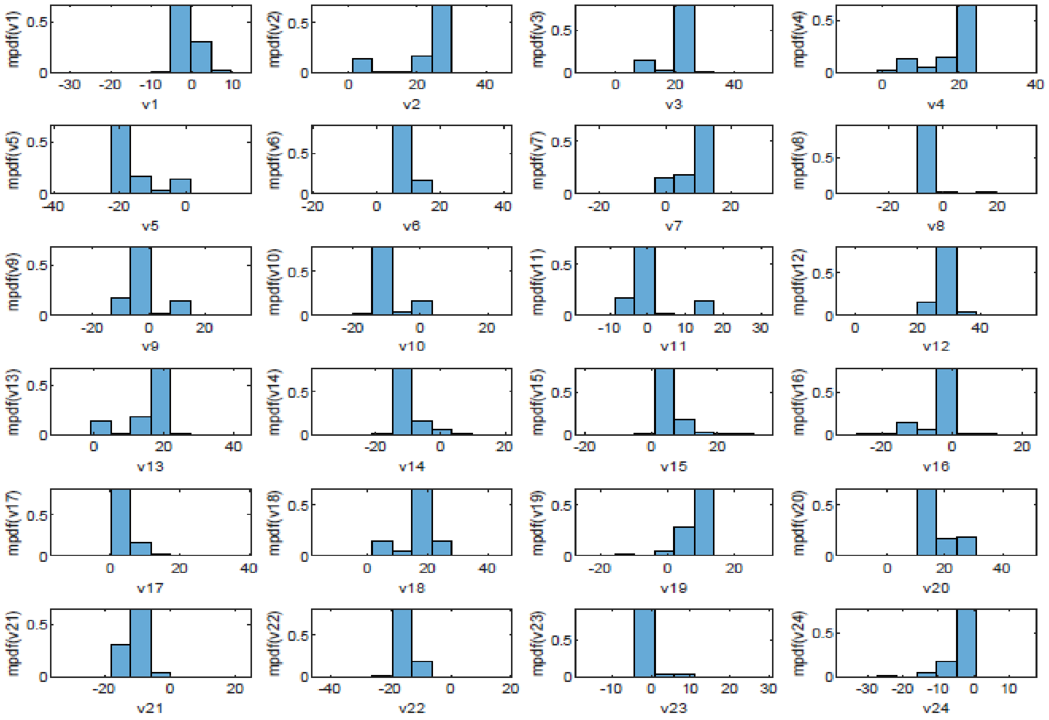


(continued)

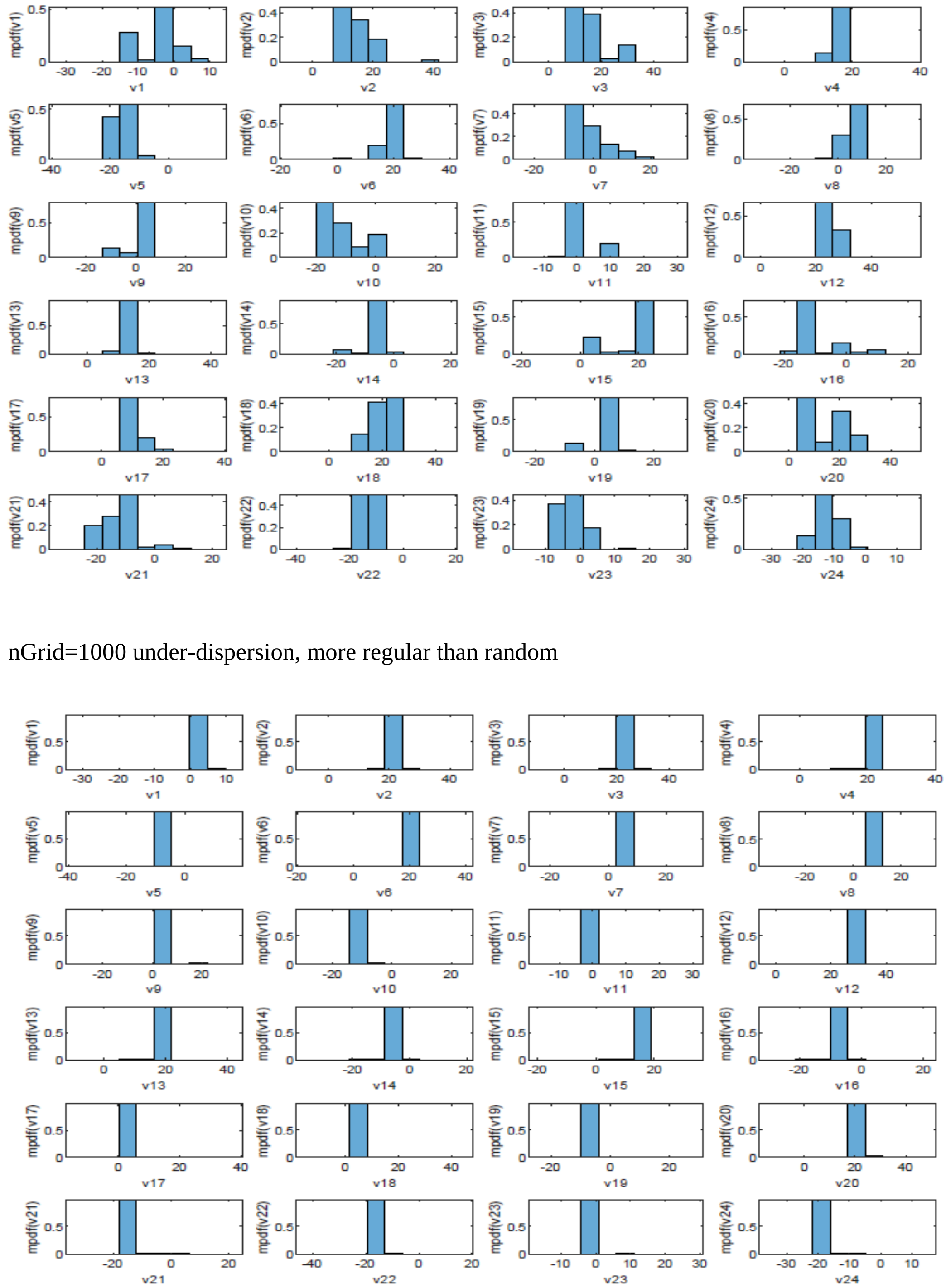
(c) nGrid=10



(d) nGrid=20



(continued)


(e) nGrid=100

(f) nGrid=1000 under-dispersion, more regular than random

Figure A3. Marginal distributions of the fluxes of Core Metabolic Network of *Escherichia coli* under different scenarios by the DISCOPOLIS algorithm with 10,000 random samples taken and 10 random seeds.