A Review of Population Initialization Techniques for Evolutionary Algorithms

Borhan Kazimipour
Xiaodong Li
A.K. Qin

RMIT UNIVERSITY

Outlines

1. **Introduction**
2. **Categorization**
3. **Randomness**
4. **Compositionality**
5. **Generality**
6. **Discussions**
7. **Conclusion**

Outlines

1. **Introduction**
2. Categorization
3. Randomness
4. Compositionality
5. Generality
6. Discussions
7. Conclusion

Definition of Population Initialization

- **Definition:**
  - **Initialization** is the assignment of an initial value to a data object or variable.
  - **Population Initialization** is the assignment of newly generated or existing values as the initial location of the population members in the search space.

- **Common Parameters:**
  - Population size
  - Number of variables or dimensionality (given)
  - Variables range (given)

- **Note:** In this study our main focus is on continuous techniques capable of generating real-value numbers in continuous spaces.

Importance of Population Initialization

- **Why studying population initialization is important?**
  - **Popularity:** All population based algorithms, including EA, need a population initialization module.
    - *"initialize population randomly"* is the most widely used expression in EA community!
  - **Variety:** Lots of different population initialization techniques are proposed, so far.
    - About 80 previously published works are referenced in the paper.
  - **Effectiveness:** Clearly, starting from a *good* position makes it easier and faster to achieve the aim, than starting from a *bad* one.
    - *"Advanced initialization techniques can increase the probability of finding global optima [7], reduce the variation of the final results [13], decrease the computational costs [7] and improve the solution(s) quality [8]."*
  - **Inconsistency:** Some controversy findings have been reported.
    - *"For example, [69] claimed that the desirable effect of uniformity of initial population is more significant in high dimensions (up to 50 dimensions) while [13], in contrast, claimed that uniform initialization techniques (e.g., QRS) loose their effectiveness in problems of 12 or more dimensions."*

Outlines

1. Introduction
2. **Categorization**
3. Randomness
4. Compositionality
5. Generality
6. Discussions
7. Conclusion

Categorization of PIT

Previous Categorizations:

- There were very few works.

- They were not comprehensive.

- They categorized PITs only from one aspect

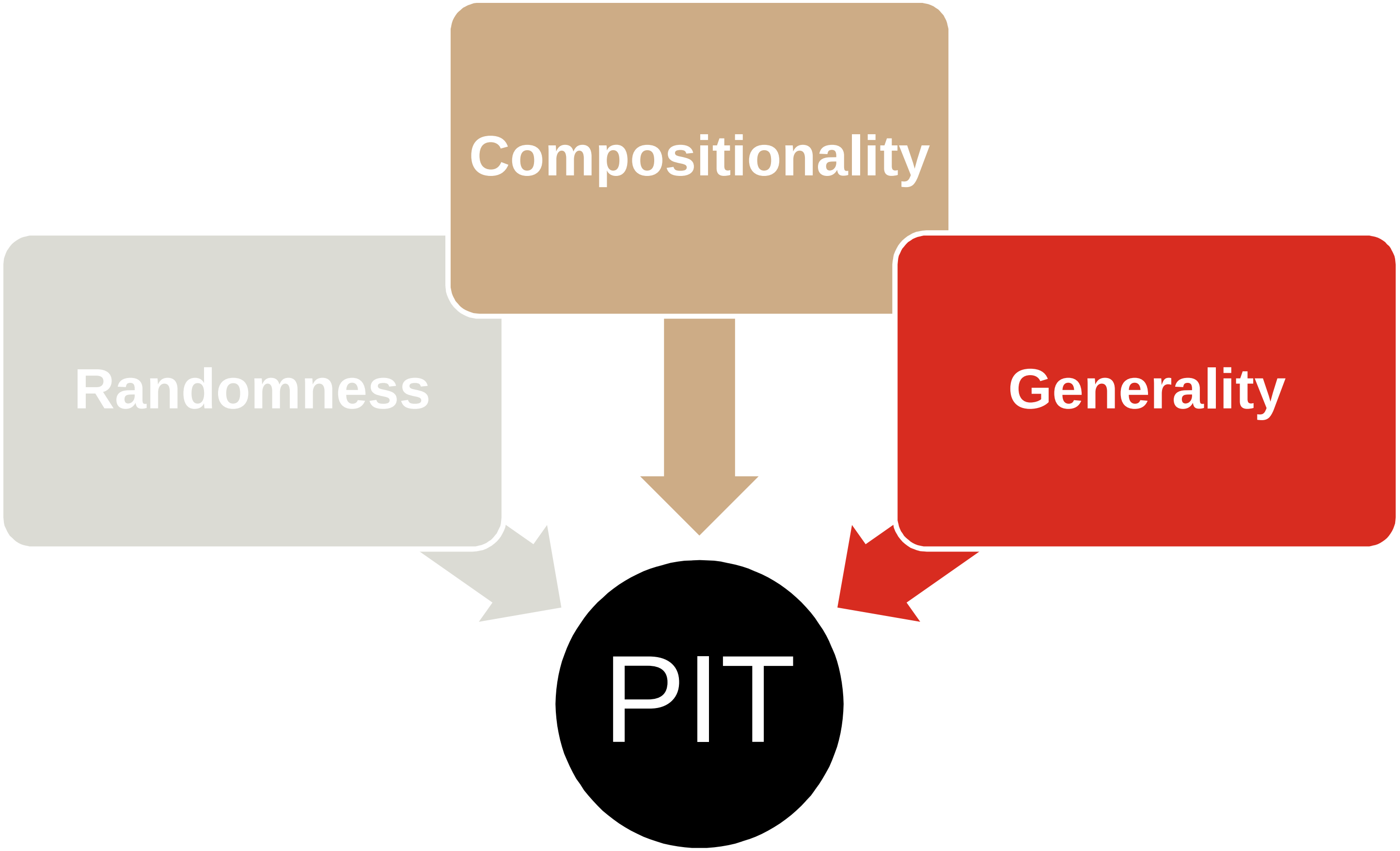
The New Categorization:

+ It covers all existing PITs

+ It groups PITs from three perspectives

+ It is clear and easy-to-understand

Categorization of PIT



Outlines

1. Introduction
2. Categorization
3. **Randomness**
4. Compositionality
5. Generality
6. Discussions
7. Conclusion

Definitions of Randomness

Completely Deterministic

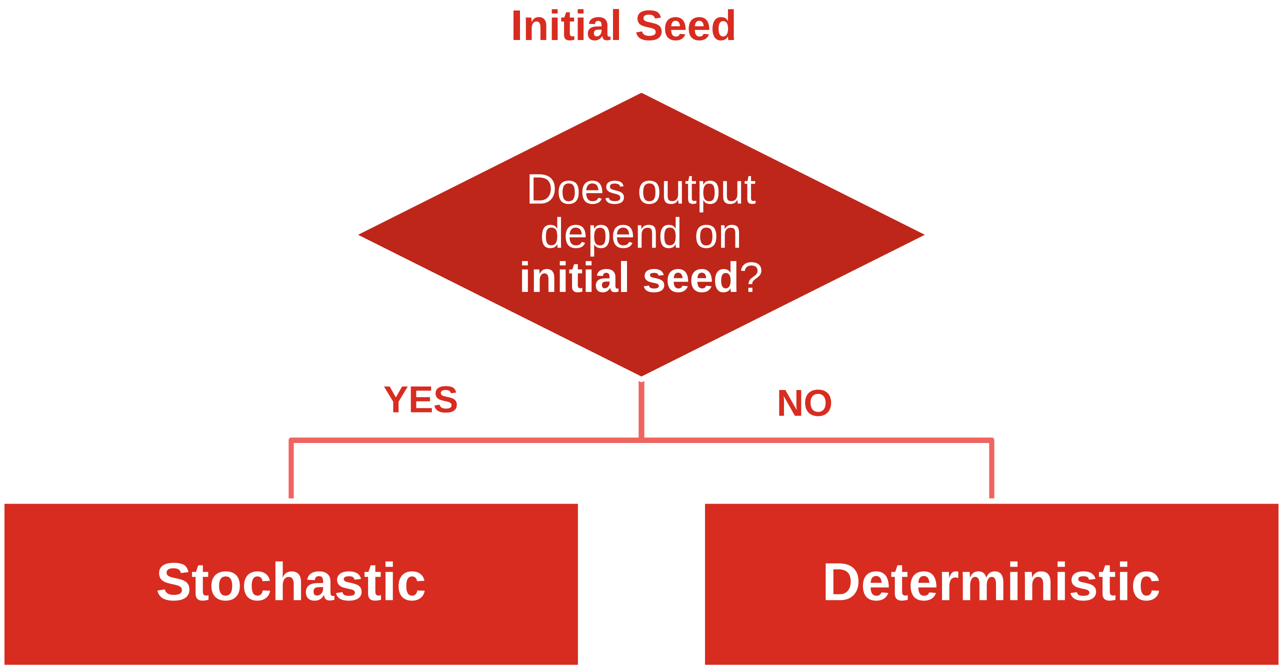
Continuum of Randomness

Truly Random

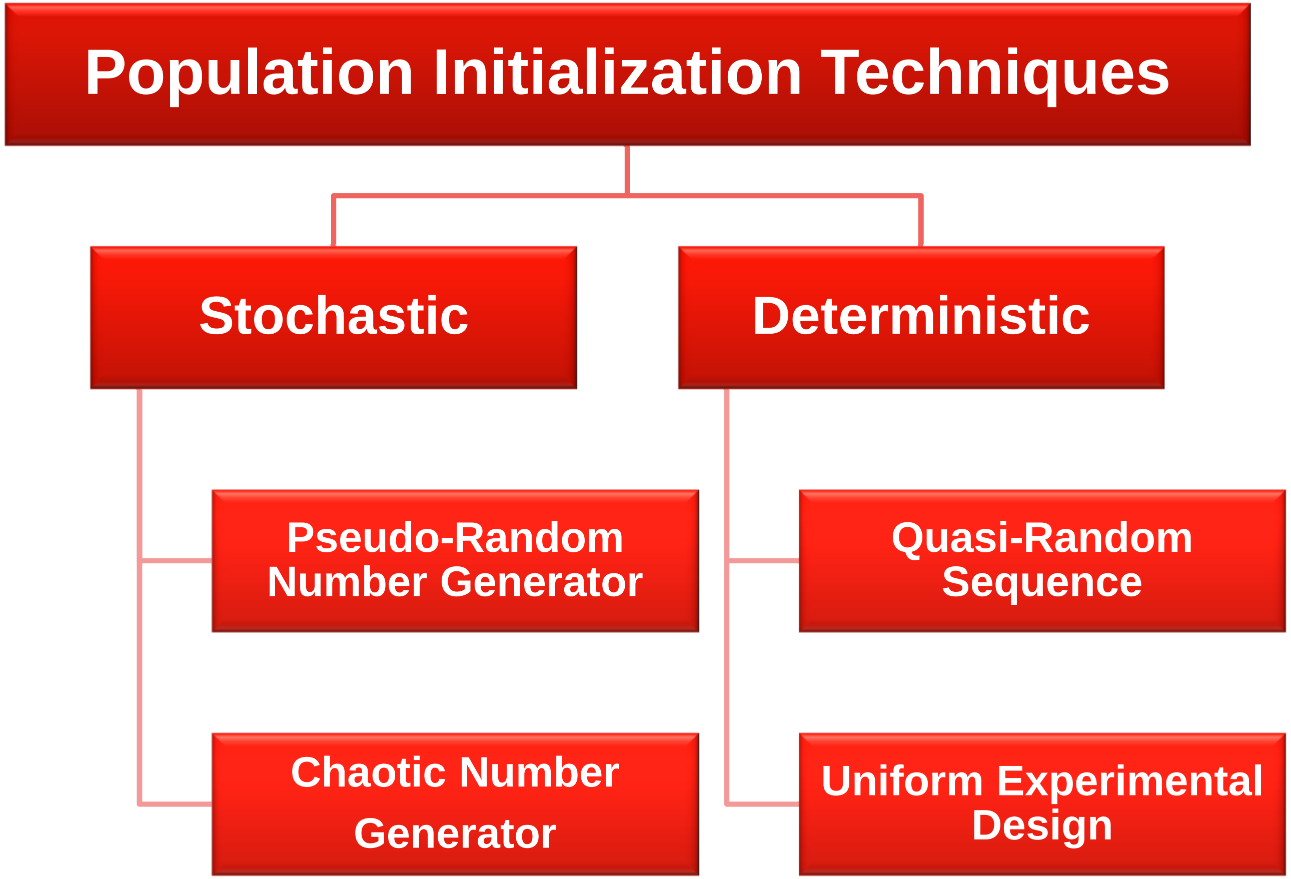
- **True Random:**
  - A true random sequence is usually described as a sequence having strong properties such as *complete unpredictability*, *incompressibility* and *irregularity*.
  - Some believe true random sequences do not exist (theoretical drawback).
  - There is no tool to proof a given sequence is truly random (emperical drawback).
- **Computational Random:**
  - A sequence is computationally random if it passes some tests on the properties of true randomness e.g., unpredictability, and incompressibility.
- **Statistical Random:**
  - A sequence is statistically random if it passes some tests on the statistical (distributional) properties of true random sequences e.g., uniformity.

Measuring Randomness

- **Issues**:
  - Categorization based on a tool is very **subjective** and cannot be generalized
  - In some cases, the tools for measuring randomness **disagree** each other.
- **Solution:**
  - Defining an easy to understand, apply and generalize criterion to differentiate PITs:



Categorization based on Randomness


Population Initialization Techniques
Stochastic
Deterministic
Pseudo-Random Number Generator
Chaotic Number Generator
Quasi-Random Sequence
Uniform Experimental Design

Stochastic PIT

- **Definition:**
  - Population initialization techniques where their results depend on initial seeds, are labelled as stochastic initializers.
- **Property:**
  - Unpredictable (computationally)
  - Irregularity
- **Subcategories:**
  - Pseudo-Random Number Generator (PRNG)
  - Chaotic Number Generator (CNG)

Pseudo-Random Number Generator (PRNG)

- **Why we use PRNGs:**
  - The disability of deterministic machines (i.e., digital computers) in producing true random numbers
  - The lack of efficient techniques to sample random numbers from physical phenomena e.g., radioactive decay or atmospheric noise.
- **Properties:**
  - PRNGs can be ranked based on
    - **Cycle time** or **period length:** the smallest integer that a PRNG repeats producing previously produced numbers
    - **Equidistribution:** all points in the range have equal frequency or probability of occurrence..
- **Examples:**
  - WELL, KISS, and Mersenne Twister (Matlab rand function)
- **Tests:**
  - DieHard and TestU01

Pseudo-Random Number Generator (PRNG)

- **Advantages:**
  - **Popularity:** PRNGs are the most commonly used population initializers in EAs
  - **Simplicity:** <u>Fast</u> PRNG tools are available in every programming language
  - **Limitless:** There is no restriction on the number of points (i.e., <u>population size</u>) and number of decision variables (i.e., <u>dimension size</u>)
  - **Uniformity:** not very high and population size is large enough, PRNGs can provide initial populations with satisfactory level of uniformity.
  - **Variety of distribution:** Uniform populations generated by PRNGs can be easily transformed to biased populations (not evenly distributed). Gaussian distribution

- **Disadvantages:**
  - curse of dimensionality

- **Improvements:**
  - Increasing population size for high dimensional problems, when possible.

Chaotic Number Generator (CRNG)

- **Introduction**
  - Chaos theory studies the behaviour of dynamical systems which are very sensitive to their initial conditions.
- **Definition**
  - To generate chaotic numbers, a proper formula (chaotic map) should be run, iteratively.
  - General form of one-dimensional chaotic maps $x^{k+1} = f_m(x_k)$, while x^0 is the random initial seed.
- **Properties**
  - **Ergodicity** The ability to traverse all states in a certain region.
  - **Higher Order Regularity**

Chaotic Number Generator (CNG)

Example

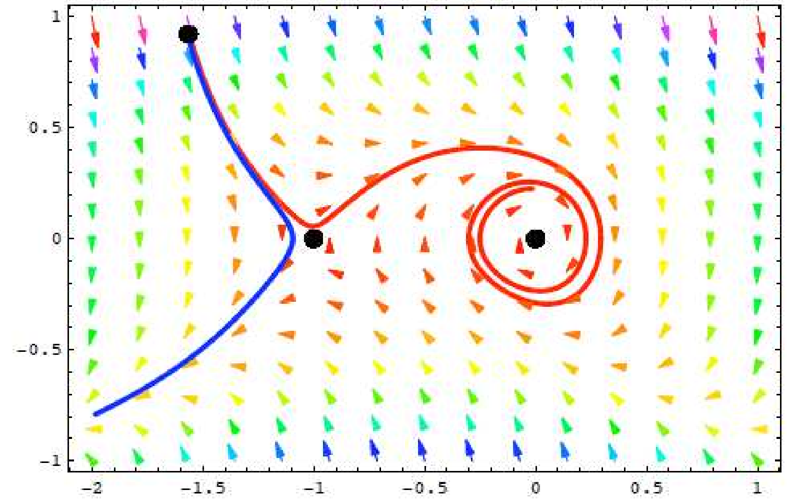


Fig. 8 State space trajectory for a dynamic system with 2 singular points s_1 and s_2. On the position $s_1 = \{0,0\}$ is repellor and at the position $s_2 = \{-1,0\}$ saddle. Start points of both trajectories diverge despite fact that this coordinates (x_1 = {-1.56, 0.92} and x_2 = {-1.57, 0.92}) are very close.

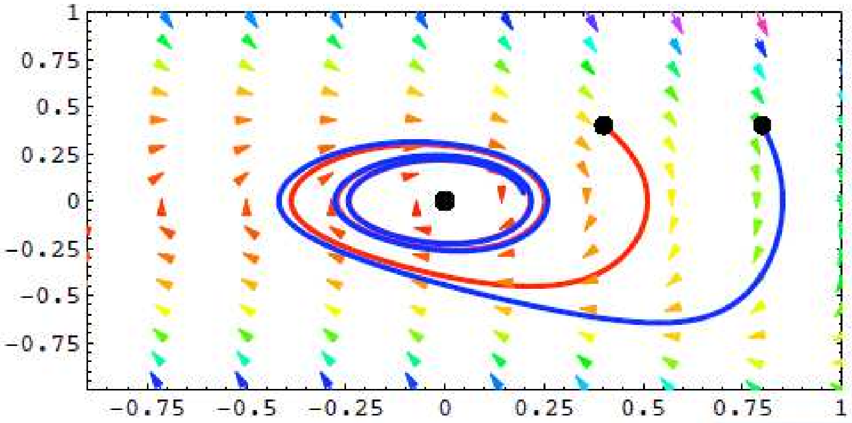


Fig. 9 Different behavior can be observed when both trajectories will start in different part of state space. Despite its bigger difference in starting possition (x_1 = {0.4, 0.4} and x_2 = {0.8, 0.4}) trajectories merge together after certain time.

- Mutual Relations of Evolutionary Dynamics, Deterministic Chaos and Complexity, Prof. Ivan Zelinka, CEC 2013

Chaotic Number Generator (CNG)

- **Advantages:**
  - It has been shown that adopting CNGs can improve performance of EAs in terms of
    - population diversity
    - success rate
    - convergence speed
- **Disadvantages:**
  - It is not clear which one produce more uniform population; PRNG or CNG.
  - Most of previously proposed chaotic maps are designed for one, two or three dimensional spaces.
  - the behaviour of CNGs are very sensitive to the initial condition, parameter settings, and precision of their implementations.
  - Existence of some attractors may cause the population to converge to a few fixed points.
  - It is not clear yet why in some cases a few maps perform considerably better than the other maps.
  - General practitioners may face difficulties finding the best maps and parameter configuration for their particular problems.

Deterministic PIT

- **Definition:**
  - Techniques which always generate the same population regardless of the initial seed are known as deterministic techniques.
- **Property:**
  - In contrast with the stochastic techniques, uniformity is more important than randomness or unpredictability.

In the **absence of prior knowledge** about the problem, **uniformity** of the initial population can enhance the *exploration* ability of EAs in early iterations.*

- **Advantages**:
  - Converging to a better solutions in terms of objective values
  - Saving a considerable amount of computational budget

* some researchers doubt about this belief!

Discrepancy

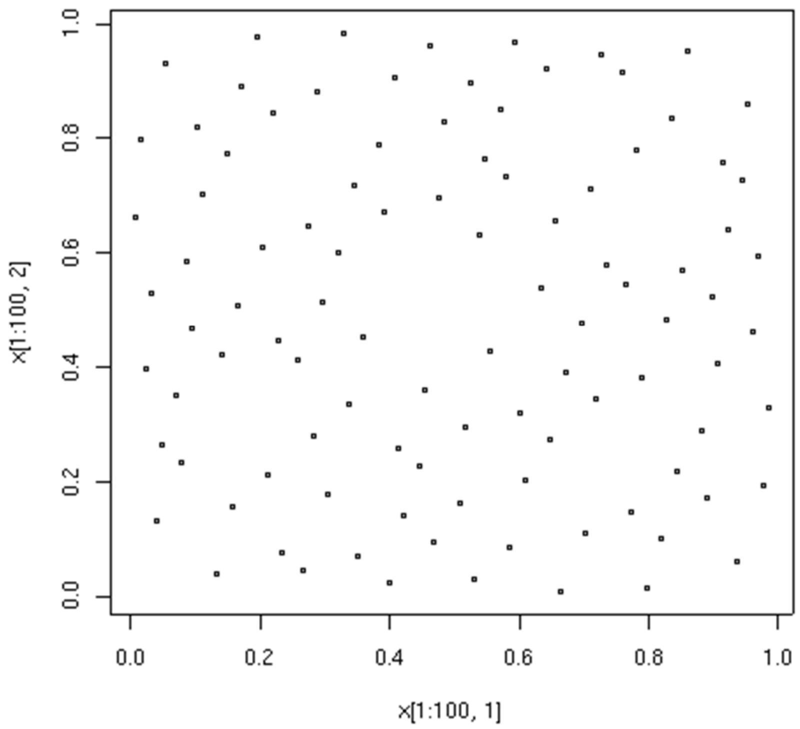
- **Discrepancy:**
  - Literally, discrepancy means *non-uniformity.*
  - Technically, discrepancy measures are tools for determining non-uniformity level of a given point set.
    - Star discrepancy
    - Central discrepancy
- **Low-Discrepancy:**
  - Point sets with low discrepancy are those with <u>high level of uniformity</u>.
  - low-discrepancy point generator is another name for deterministic number generators
- **Subcategories:**
  - Quasi-Random Sequence (QRS)
  - Uniform Experimental Design (UED)

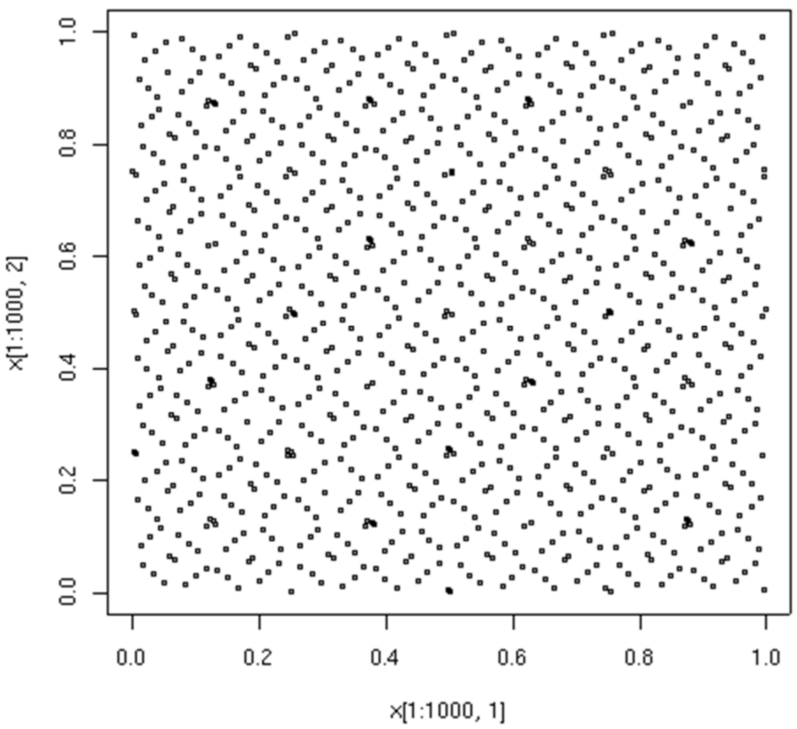
Quasi-Random Sequence (QRS)

- **Introduction:**
  - QRSs are completely deterministic and no random element involved.
  - QRSs are very regular and uniform
- **Property:**
  - Having theoretical upper-bounds on discrepancy.
  - QRSs try to find the optimal parameters to decrease the upper-bounds or to approach the lower-bounds
- **Advantages:**
  - Uniform population when population size is big enough.
  - Before running EA, the best QRS can be selected based the theoretical upper-bound on non-uniformity.
- **Disadvantages:**
  - The bounds may not be very beneficial in practice (small sample size in high dim.)
  - Discrepancy measures in some cases contradict each other.
  - Any correlation between discrepancies and solution’s objective values has not been proven yet.

Quasi-Random Sequence (QRS)

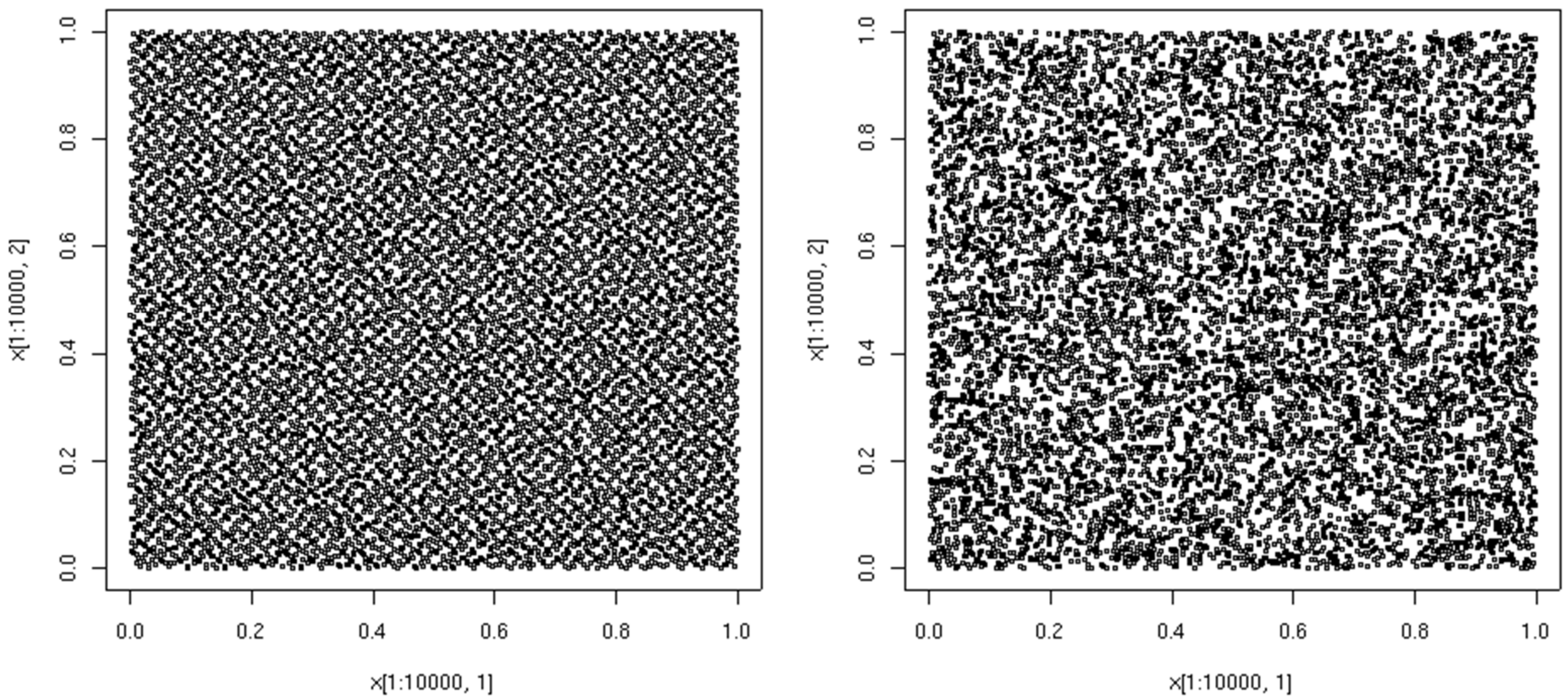
Example I





Quasi-Random Sequence (QRS)

Example II



Uniform Experimental Design (UED)

- **Introduction:**
  - UEDs are space-filling algorithms which look for points to be evenly scattered in a given range.
  - Suppose we seek a **complete grid** in a *D* dimensional space which each variable has exactly *q* different values (i.e., levels). Then, the total number of points in the grid (i.e., population size) would be q^D.
  - UEDs can be used to systematically select a smaller number of points from the complete grid which is still uniform.

- **Examples:**
  - Uniform Design
  - Orthogonal Design

Uniform Experimental Design (UED)

- **Advantages:**
  - UEDs not only consider one-dimensional uniformity (like QRSs), but also two and *D*-dimensional projection uniformities (result in more regularity).
  - UEDs can generate both discrete points (ideal for nominal and discrete optimization problems) and real-value numbers.
- **Disadvantages:**
  - The performance of UEDs depends on the **parameter settings** (very poor results if parameters are set improperly).
  - **Computationally** not applicable in high-dimension problems.
    - Solution I: subset selection
    - Solution II: variable clustering

Outlines

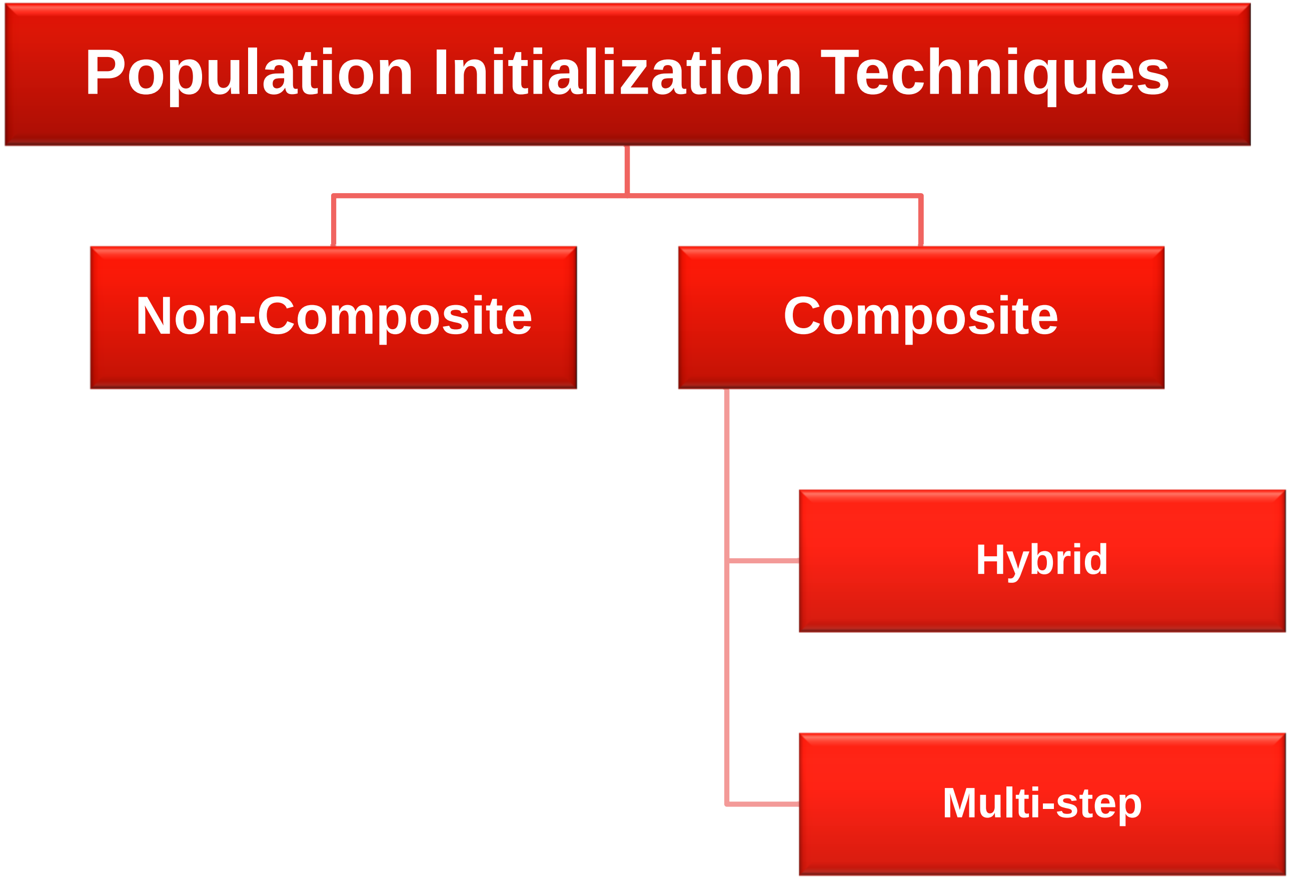
1. Introduction
2. Categorization
3. Randomness
4. **Compositionality**
5. Generality
6. Discussions
7. Conclusion

Compositionality

- **Definition:**
  - The number of standalone procedures that are involved in PITs is used as categorization criterion.
- **Categorization:**
  - According to this criterion, PITs fall into **composite** and **non-composite** groups.

- **Non-composite:**
  - **Definition:** All stand alone PITs
  - **Examples:** PRNG, CNG, QRS, UED…
- **Composite:**
  - **Definition:** PITs which are built based on two or more components. The building component may or may not be a non-composite PIT.
  - **Examples:** Using a PRNG to produce an initial seed of a CNG, or vice versa.

Categorization based on Compositionality



Hybrid PIT

- **Definition:**
  - A hybrid PIT comprises of several components which all of them can be used as standalone PITs.
- **Examples:**
  - Using a PRNG to produce an initial seed of a CNG, or vice versa.
  - Hybridizing stochastic PITs with deterministic PITs to have uniformity and randomness at the same time: random start QRS, scrambled QRS and mixed pseudo-quasi-random sequence.
- **Advantages:**
  - Hybrid PITs may inherit the advantages of the basic techniques which they are made from.
- **Disadvantages:**
  - Hybrid PITs may inherit the disadvantages of the basic techniques which they are made from.

Multi-step PIT

- **Definition:**
  - Multi-step PITs comprise of two or more components which at least one of them cannot be employed as an standalone PIT.
  - Multi-step techniques generally process and refine the previously generated population in later steps.
- **Examples:**
  - **Fitness:**
    - Opposition Based Learning: quasi-reflection opposition-based learning, centre-based sampling, generalized opposition-based learning and current optimum opposition-based learning,…
    - local and global selections, hill-climbing local search, Tabu search, …
    - quadratic interpolation
    - non-linear local search (simplex)
    - smart sampling
  - **Uniformity:**
    - Centroidal Voronoi Tessellation (CVT)
    - Simple Sequential Inhibition Process (SSI)

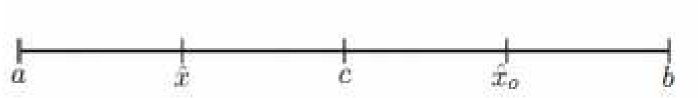
Multi-step PIT

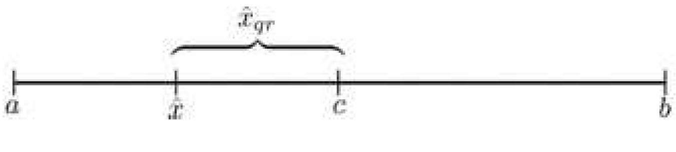
Opposition Based Learning

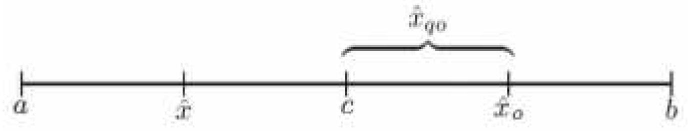
- **Procedure:**
  - Step I: Generate a set of points called original population using any arbitrary PIT.
  - Step II: Apply simple heuristic rules to produce another population with the same size and call it opposition population.

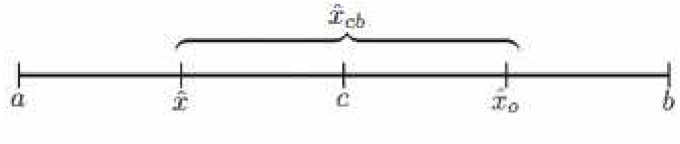
$$\tilde{x}_{i,j} = a_j + b_j - x_{i,j}, \quad j = 1, ..., D.$$

  - Step III: Merge two populations.
  - Step IV: Select the best subpopulation based on fitness values of the individuals.
- **Examples:**









Multi-step PIT

Opposition Based Learning (OBL)

- **Advantages:**
  - There is 50% chance that an unknown solution is closer to the opposition point than the original point (theoretically proved).
  - In some variation of OBL, this chance is even higher (theoretically proved).
- **Disadvantages** (the same for all multi-step PITs exploiting objective function)**:**
  - They consume a part of computation budget to evaluate the fitness function and select the best subset of the population.
  - They refine the final population based on the very fist population (sensitive to the performance of the PIT they employed in the first step).
  - Because of the greedy selection mechanism, the chance of losing informative building blocks is very high. (it is very probable that individuals which have useful subcomponents are immediately excluded only due to their low fitness values in comparison with the other individuals in initialization step).

Multi-step PIT

Centroidal Voronoi Tessellation (CVT)

- **Properties:**
  - Instead of fitness function, CVT uses other metrics to enhance initial population quality.
- **Procedure:**
  - Step I: Generate a set of points called original population using any arbitrary PIT.
  - Step II: By the aid of many randomly generated auxiliary points, divide search space into some partitions.
  - Step III: Adjust the partition centroids to have a more evenly distributed partitions.
  - Step IV: Do steps II and III until some criteria met.
  - Step V: Use partition centres as the initial population of EA.

Multi-step PIT

Centroidal Voronoi Tessellation (CVT)

- **Advantages** (similar to all multi-step PITs not exploiting objective function)**:**
  - They are able to produce geometrically uniform populations (without using any objective function evaluations).
  - Since they do not select pints based on fitness values, it is less likely to miss a great part of search space as greedy selection.
- **Disadvantages:**
  - Computationally expensive (but can be hybridize with more uniform PITs)
  - Their performance depends on the internal partitioning (or clustering) algorithm or employed distance measures.
  - These iterative techniques might not converge when the population size is relatively small (high dimension).

Outlines

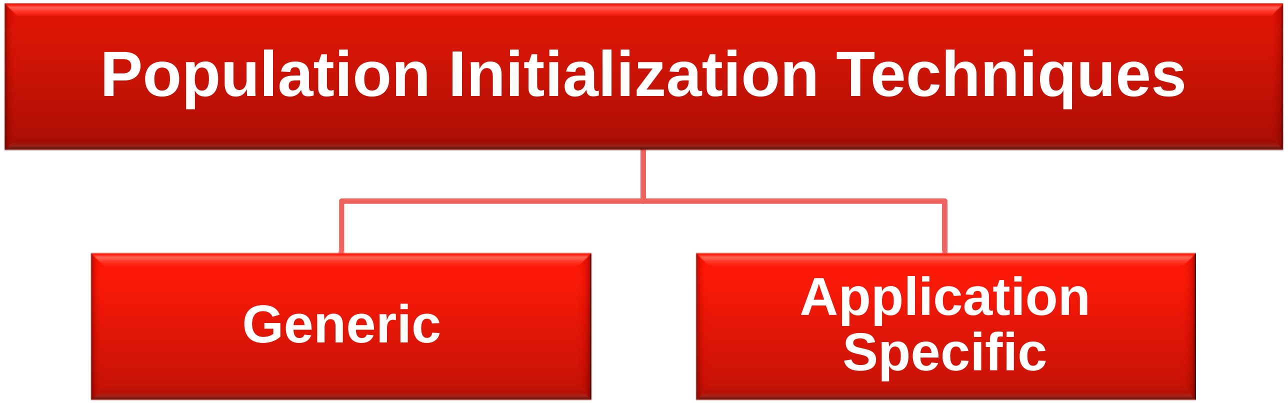
1. Introduction
2. Categorization
3. Randomness
4. Compositionality
5. **Generality**
6. Discussions
7. Conclusion

Generality

- **Definition:**
  - Generality of PITs refers to the variety of the domains that it can be applied to..
- **Categorization:**
  - According to this criterion, PITs fall into **generic** and **application specific** groups.

- **Generic:**
  - **Definition:** All PITs which can be directly applied to all types of optimization problems are called generic techniques.
  - **Examples:** All PITs described in previous slides.
  - **Advantages:** No assumptions about the problem (ideal for black-box optimization)
- **Application Specific:**
  - **Definition:** PITs which are specially designed to be applied to particular real world problems.
  - **Advantages:** Using expert/domain knowledge.
  - **Disadvantages:** Cannot be applied on a variety of other problems.

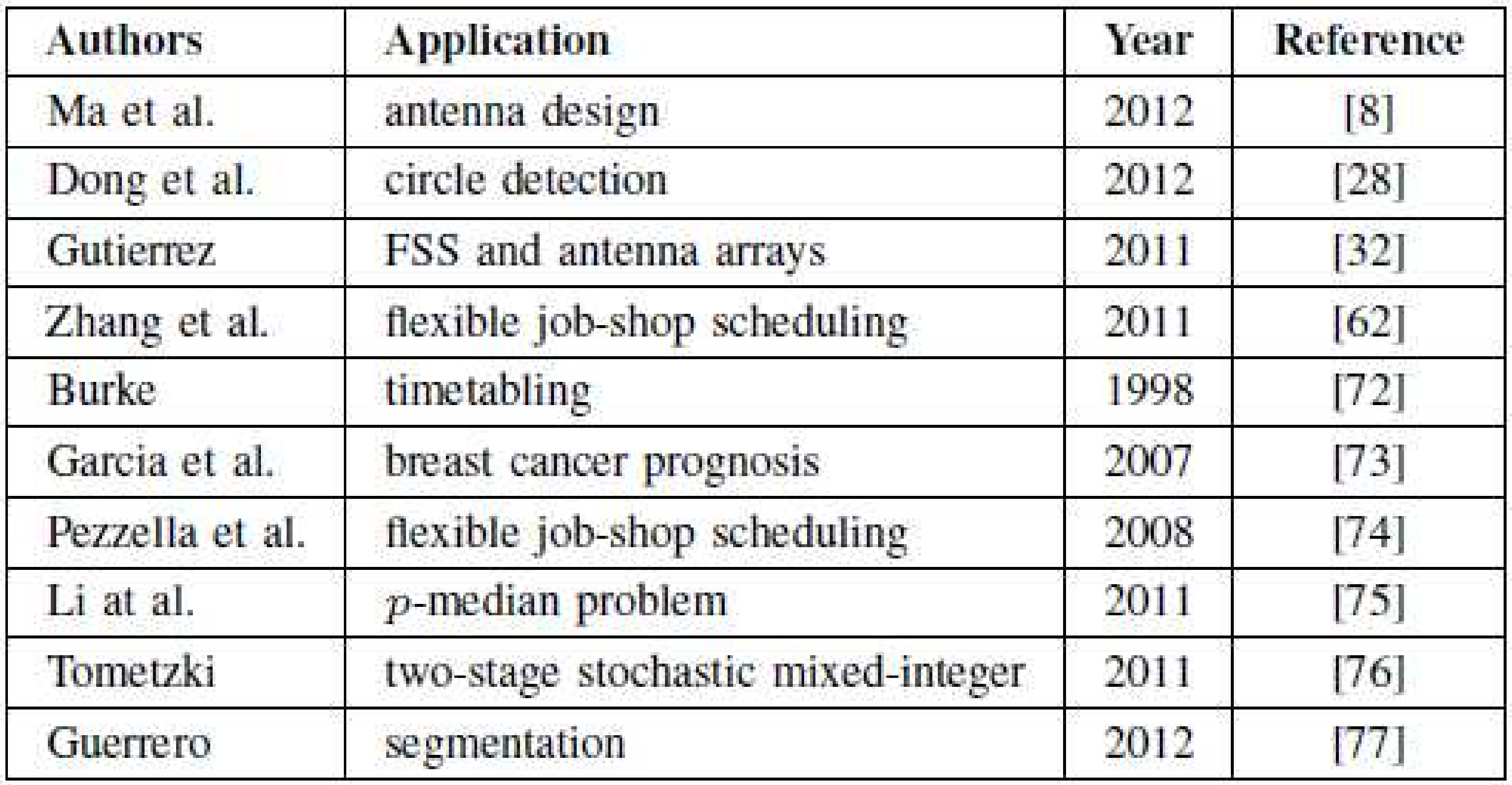
Categorization based on Generality



Generality

Examples of application specific PITs

TABLE I. Application Specific Techniques



Authors	Application	Year	Reference
Ma et al.	antenna design	2012	[8]
Dong et al.	circle detection	2012	[28]
Gutierrez	FSS and antenna arrays	2011	[32]
Zhang et al.	flexible job-shop scheduling	2011	[62]
Burke	timetabling	1998	[72]
Garcia et al.	breast cancer prognosis	2007	[73]
Pezzella et al.	flexible job-shop scheduling	2008	[74]
Li at al.	p-median problem	2011	[75]
Tometzki	two-stage stochastic mixed-integer	2011	[76]
Guerrero	segmentation	2012	[77]

Outlines

1. Introduction
2. Categorization
3. Randomness
4. Compositionality
5. Generality
6. **Discussions**
7. Conclusion

Discussions

- **Dimensionality**
  - Nearly all previous studies, have been done on low dimensional single objective problems (less than 60 dimensions).
  - Some studies on low dimensions tried to generalize their findings to higher dimensions. However, there has been little agreement on validation of those findings in high dimensional spaces.
- **Few Comparisons:**
  - Most comparison studies on PITs are limited to a few (mostly less than four) techniques.
  - In many cases, the techniques are selected arbitrarily and without considering similarities and dissimilarities of the selected techniques and their categories (lack of a comprehensive categorization).

Discussions

- **Effective Parameters:**
  - The relationship between population initialization and other parameters are almost completely neglected in previous works.
    - mutual effect of population size,
    - computational budget,
    - exploration/exploitation ability of the algorithm
    - the characteristics of the underlying problems

- **Real World Problems:**
  - The effect of different population initialization techniques on real-world problems are not explored enough.
  - Mostly tested on benchmarks

Discussions

- **Practical Rules/Suggestions**
  - Beside some theoretical advices (like this study), too few practical rules of thumb are provided for choosing proper PITs according to different situation.
  - From a practitioners point of view, it is still unclear which PIT matches a specific EA model or suits to a given optimization problem.

- **Multi and Many Objective Problems:**
  - Too few studies tried to apply and investigate the potential application of advanced initialization techniques on multi and many objective optimization problems.

Outlines

1. Introduction
2. Categorization
3. Randomness
4. Compositionality
5. Generality
6. Discussions
7. **Conclusion**

Conclusion

- The volume of the surveyed techniques reveals that population initialization has become an **active research topic** in the EA domain.
- Initialization is an important task in EA which **worth further studies**.
- The provided categorization benefits researchers in choosing proper **state-of-the-art** PITs for their research.
- The provided categorization benefits practitioners to find proper PITs for their particular **applications**.

Running to 203A for Next Presentation @ 9:50

Effects of Population Initialization on Differential Evolution for Large Scale Optimization

Thank you

☺