SUSTech
Southern University of Science and Technology

Scalable Evolutionary Search

Ke TANG

Shenzhen Key Laboratory of Computational Intelligence
Department of Computer Science and Engineering
Southern University of Science and Technology (SUSTech)
Email: tangk3@sustc.edu.cn

November 2018 @ CSBSE, BUCT

Outline

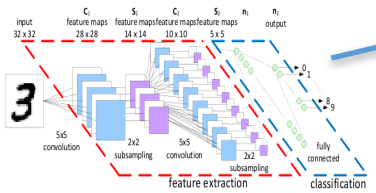
- **Introduction**
- General Ideas and Methodologies
- Case Studies
- Summary and Discussion

Introduction

- Evolutionary Algorithms are powerful search methods for many hard optimization (e.g., NP-hard) problems that are intractable by off-the-shelf optimization tools (e.g., gradient descent).

Discrete Search Space

Railway timetabling



Network Optimization

Non-differentiable constraints + mixed integer search space

Truss design

Non-differentiable Objective function

Portfolio optimization

Discrete Search space

What & Why?

- It is important to make EAs scalable
  - Scalability plays a central role in computer science.
  - Scalability is more important than ever when employing EAs to tackle hard problems of ever growing size.

- Scalability describes the relationship between some environmental factors and the measured qualities (e.g., runtime or solution quality) of systems/software/algorithms.

- Environmental factors
  - Decision variables
  - Data
  - Computing facilities, e.g., CPUs
  - etc.

What & Why?

- Take the rise of big data as an example, which brings huge challenge to evolutionary search.

- Data: $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\} \subset \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$

- Goal: minimize the generalization error

Huge No. of model parameters

$$\mathcal{E}(h) := \mathbb{E}_{\mathbf{z}}\big[c(\mathbf{x})\ell(y, h(\mathbf{x}))\big] = \int_{\mathcal{Z}} c(\mathbf{x})\ell(y, h(\mathbf{x}))d\rho(\mathbf{z})$$

Huge volume of data

Outline

- Introduction
- **General Ideas and Methodologies**
- Case Studies
- Summary and Discussion

Scalable w.r.t. Decision Variables

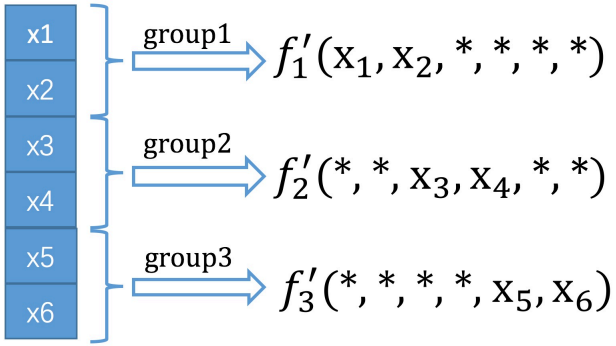
Suppose we have an optimization problem:

$$\text{minimize} \quad f(x_1, x_2, \ldots, x_D)$$

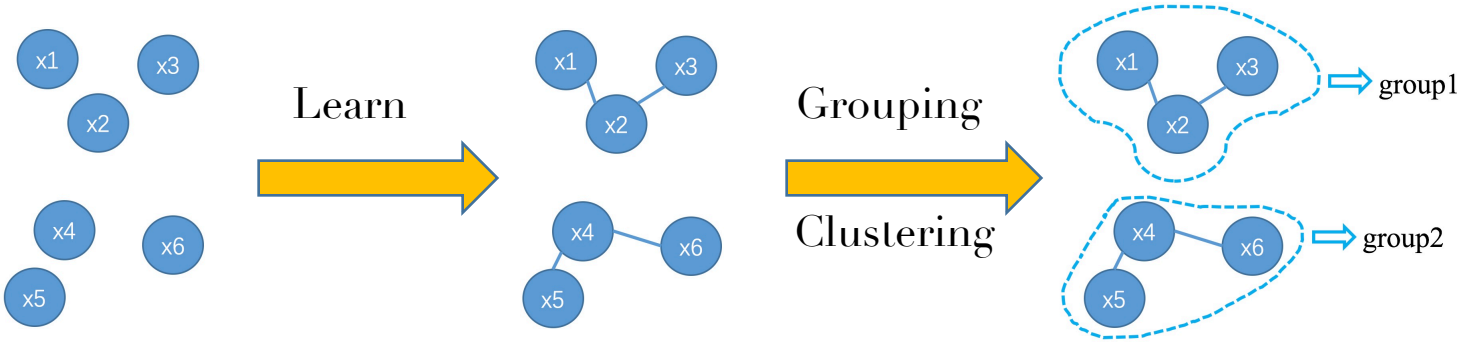
How to cope with the search space that increases rapidly with *D*?

Scalable w.r.t. Decision Variables

- Basic idea: Divide-and-Conquer



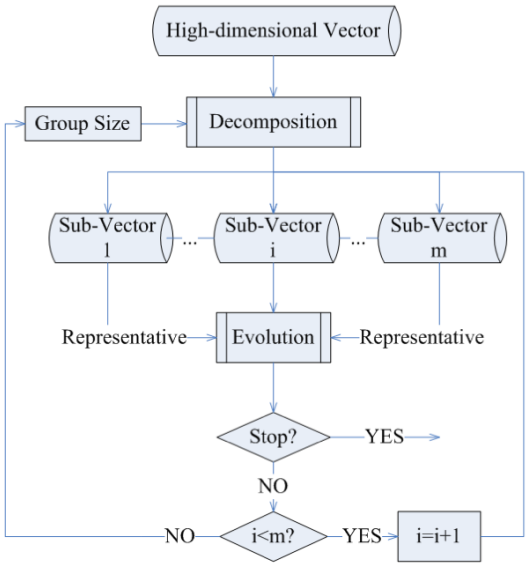
- Challenge: little prior knowledge about
  - whether the objective function is separable at all.
  - how the decision variables could be divided: Randomly or Learn to Group



Scalable w.r.t. Decision Variables

- The sub-problems can be tackled independently, but it'd be better to correlate the solving phases, because:
  - The learned relationships between variables are seldom perfect.
  - Sometimes the problem itself is not separable at all.

- A natural implementation: Cooperative Coevolution

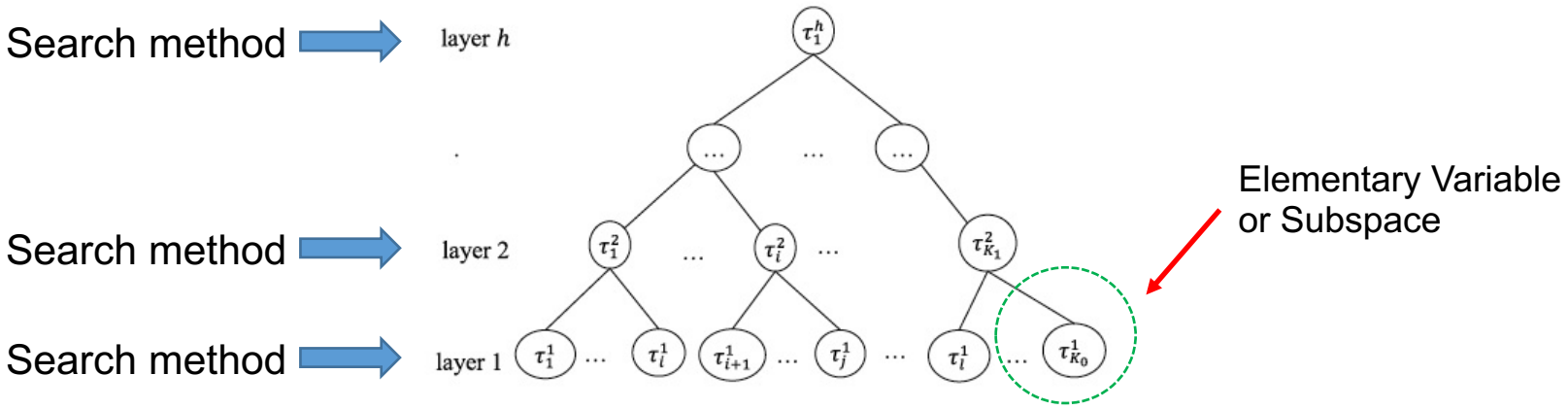


Z. Yang, K. Tang and X. Yao, "Large Scale Evolutionary Optimization Using Cooperative Coevolution," *Information Sciences,* 178(15): 2985-2999, August 2008.

W. Chen, T. Weise, Z. Yang and K. Tang, "Large-Scale Global Optimization using Cooperative Coevolution with Variable Interaction Learning," in *Proceedings of PPSN2010*.

Scalable w.r.t. Decision Variables

- CC-based methods divide a problem in a "linear" way, e.g., divide *D* variables into *K* groups of size *D*/*K*.
- The conflict between *K* and *D*/*K* restricts the application of CC.
- Remedy: build hierarchical structure (e.g., tree).
  - Different layers re-defines the solution space with different granularity.
  - "Applying a search method to different layers" ~ "search with different step-sizes"



Scalable w.r.t. Decision Variables

What About Multi-Objective Optimization?

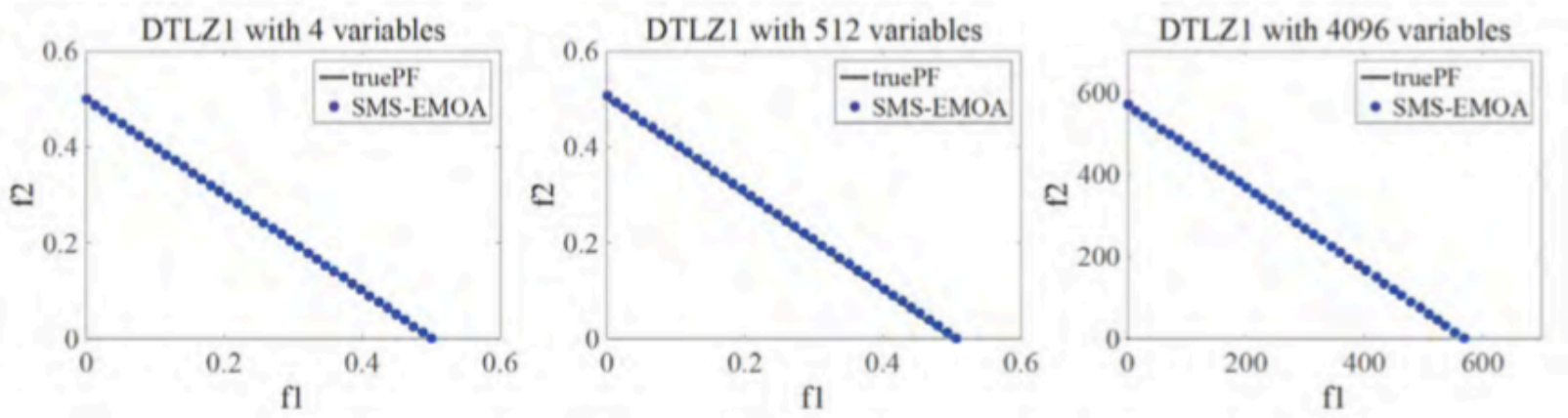
Scalable w.r.t. Decision Variables

- Are all MOPs difficult?
- Why an MOP is difficult (in comparison to an SOP)?

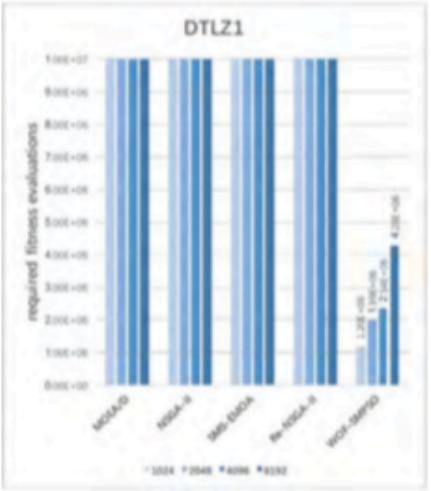
Table VII The efficiency of the algorithms on ZDT problems, with mean and standard deviation. A slash '–' indicates that the algorithm cannot achieve the required quality in all 25 runs. The best performance validated by the Wilcoxon Rank Sum is highlighted in grey.

Problem	D	MOEA/D		NSGA-II		SMS-EMOA		DW-MOEA	
ZDT1	1024	5.14E+05	1.52E+04	3.45E+05	4.82E+03	2.19E+05	3.66E+03	2.05E+05	3.28E+03
ZDT1	2048	1.16E+06	2.18E+04	7.59E+05	2.29E+04	4.74E+05	4.86E+03	4.10E+05	1.29E+03
ZDT1	4096	2.69E+06	1.89E+04	1.71E+06	3.15E+04	1.04E+06	1.37E+04	8.19E+05	8.96E+03
ZDT1	8192	6.54E+06	5.33E+04	3.74E+06	1.52E+04	2.24E+06	1.68E+04	1.64E+06	1.02E+04
ZDT2	1024	2.56E+05	8.11E+04	3.76E+05	1.41E+04	2.56E+05	5.47E+03	2.86E+05	2.36E+04
ZDT2	2048	6.69E+05	1.16E+05	8.35E+05	1.11E+04	5.43E+05	6.01E+03	5.71E+05	4.19E+04
ZDT2	4096	1.80E+06	6.09E+03	1.86E+06	5.29E+03	1.18E+06	7.50E+03	1.15E+06	1.89E+04
ZDT2	8192	3.94E+06	2.45E+04	4.07E+06	3.30E+04	2.51E+06	1.23E+04	2.29E+06	2.78E+04
ZDT3	1024	5.43E+05	1.66E+04	2.90E+05	6.29E+03	2.25E+05	7.15E+03	2.46E+05	4.10E+04
ZDT3	2048	1.27E+06	8.77E+03	6.28E+05	1.30E+04	4.97E+05	4.74E+03	4.10E+05	8.19E+04
ZDT3	4096	2.95E+06	2.72E+04	1.36E+06	1.80E+04	1.07E+06	1.16E+04	8.19E+05	5.46E+04
ZDT3	8192	6.72E+06	2.49E+04	2.89E+06	2.17E+04	2.33E+06	1.04E+04	1.64E+06	3.92E+04
ZDT4	1024	–		8.43E+06	2.77E+05	5.87E+06	1.54E+05	3.66E+06	2.36E+04
ZDT4	2048	–		–		–		9.61E+06	2.88E+05
ZDT4	4096	–		–		–		–	
ZDT4	8192	–		–		–		–	
ZDT6	1024	1.45E+06	1.03E+04	1.24E+06	1.36E+04	9.31E+05	5.57E+03	9.31E+05	2.45E+04
ZDT6	2048	3.12E+06	8.87E+03	2.69E+06	1.47E+04	1.96E+06	1.97E+04	1.83E+06	4.73E+04
ZDT6	4096	6.77E+06	1.18E+04	5.74E+06	4.32E+04	4.11E+06	2.40E+04	3.77E+06	7.37E+04
ZDT6	8192	–		–		8.59E+06	5.00E+04	7.54E+06	1.55E+05

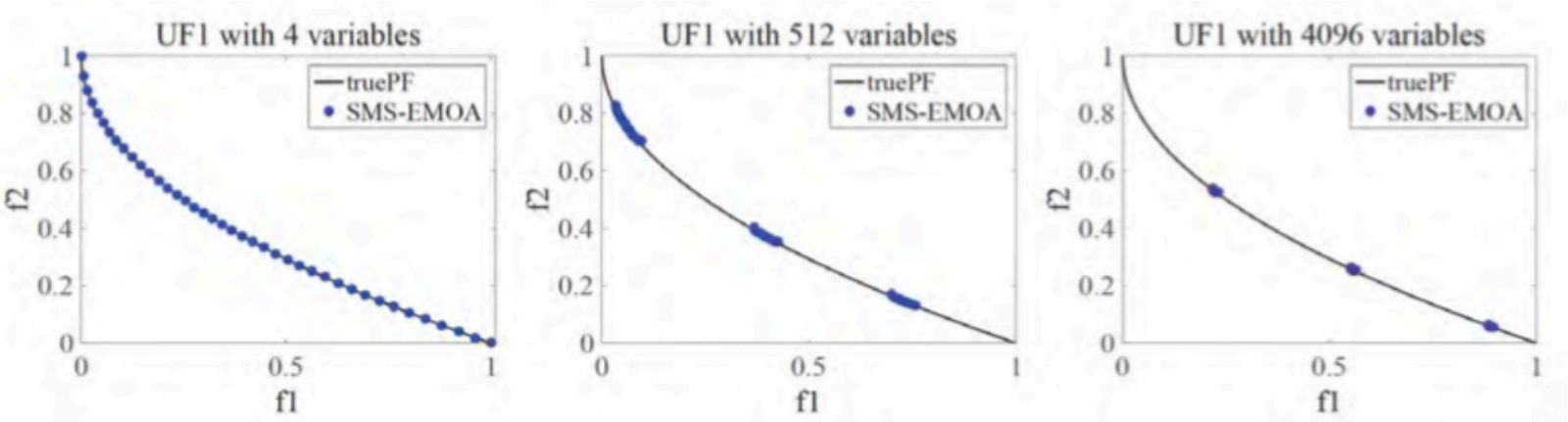
Scalable w.r.t. Decision Variables



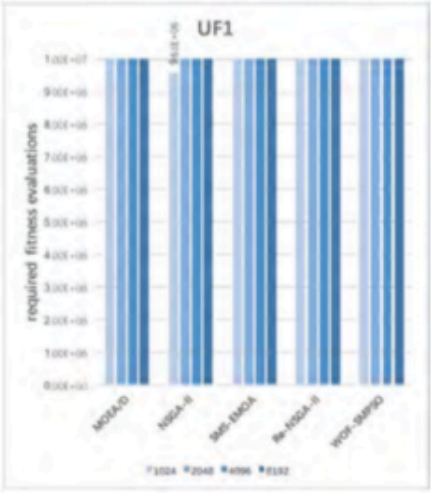
		NSGA-II	MOEA/D	SMS-EMOA	Re-NSGA-II	WOF-SMPSO
DTLZ1	1024	∗0.741	∗0.830	∗0.876	∗0.000	0.931
		0.028	0.017	0.009	0.000	0.154
DTLZ1	2048	∗0.000	∗0.000	∗0.002	∗0.000	0.912
		0.000	0.000	0.006	0.000	0.174
DTLZ1	4096	∗0.000	∗0.000	∗0.000	∗0.000	0.801
		0.000	0.000	0.000	0.000	0.233
DTLZ1	8192	∗0.000	∗0.000	∗0.000	∗0.000	0.745
		0.000	0.000	0.000	0.000	0.237



Scalable w.r.t. Decision Variables



		NSGA-II	MOEA/D	SMS-EMOA	Re-NSGA-II	WOF-SMPSO
UF1	1024	0.876	∗0.829	∗0.828	∗0.000	∗0.567
		0.019	0.042	0.034	0.000	0.007
UF1	2048	0.851	∗0.807	∗0.825	∗0.000	∗0.535
		0.011	0.036	0.017	0.000	0.030
UF1	4096	∗0.757	∗0.806	0.807	∗0.000	∗0.518
		0.091	0.037	0.029	0.000	0.045
UF1	8192	∗0.190	∗0.630	0.815	∗0.000	∗0.510
		0.159	0.062	0.012	0.000	0.041



W. Hong, K. Tang, A. Zhou, H. Ishibuchi and X. Yao, "A Scalable Indicator-Based Evolutionary Algorithm for Large-Scale Multi-Objective Optimization," ***IEEE Transactions on Evolutionary Computation***, accepted on Oct. 30, 2018.

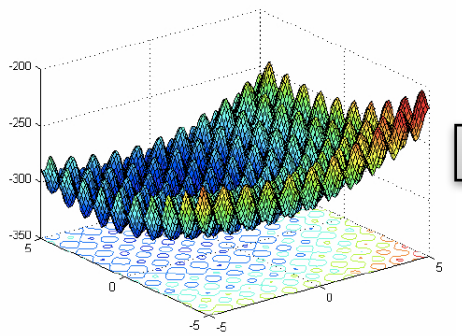
Scalable w.r.t. Processors

What can we promise if offered sufficient computing facilities for EC?

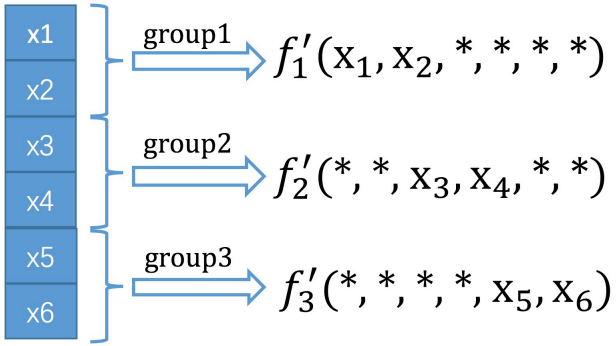
Scalable w.r.t. Processors

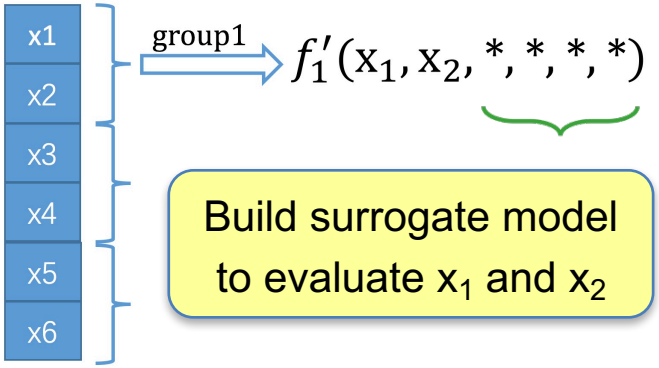
Parallel implementation of the CC approaches is nontrivial because of dependency between sub-problems.

Idea: using data generated during search course



	x_1	…	x_D	quality
datum 1	…	…	…	…
…	…	…	…	…
datum *n*	…	…	…	…





P. Yang, K. Tang* and X. Yao, "Turning High-dimensional Optimization into Computationally Expensive Optimization," ***IEEE Transactions on Evolutionary Computation***, 22(1): 143-156, February 2018.

Scalable w.r.t. Data Volume

What if the volume of data is big, while the search space is not?

Scalable w.r.t. Data Volume

- Example: tuning the hyper-parameters of Support Vector Machine
  - Only 2-3 parameters to tune.
  - Evaluating a hyper-parameter involves solving a QP, the time complexity of which is $O(n^2)$, n is the number of samples.

- Fitness evaluation using a small subset of data (like SGD)?

	Corr.	Dis.	LR	···	···	AIC.	BIC	RF.
x1	0.28	0.46	1	···	···	0.22	0.63	1
x2	0.31	0.59	0.64	···	···	0.58	0.56	1
x3	0.11	0.02	0.53	···	···	0.43	0.01	1
x4	0.1	0.1	0.64	···	···	0.73	0.92	1
x5	0.02	0.15	0.33	···	···	0.56	0.36	0.78
x6	0.36	0.02	0.01	···	···	0.32	0.02	0.22
x7	0.2	0.2	0.21	···	···	0.21	0.02	0.11
x8	0.1	0.03	0.32	···	···	0.33	0.51	0.44
x9	0.32	0.1	0.2	···	···	0.06	0.66	0
x10	0.24	0	0.02	···	···	0.6	0.03	0.33
x11	0.12	0.45	0.44	···	···	0.64	0.45	1
x12	0.36	0.58	0.12	···	···	0.73	0.58	0.67
x13	0.2	0.02	0.24	···	···	0.34	0.02	0.89
x14	0.24	0.92	0.33	···	···	0.24	0.93	0.56

regression

	Corr.	Dis.	LR	···	···	AIC.	BIC	RF.
x1	0.28	0.46	1	···	···	0.22	0.63	1
x2	0.31	0.59	0.64	···	···	0.58	0.56	1
x3	0.11	0.02	0.53	···	···	0.43	0.01	1
x4	0.1	0.1	0.64	···	···	0.73	0.92	1
x5	0.02	0.15	0.33	···	···	0.56	0.36	0.78
x6	0.36	0.02	0.01	···	···	0.32	0.02	0.22
x7	0.2	0.2	0.21	···	···	0.21	0.02	0.11
x8	0.1	0.03	0.32	···	···	0.33	0.51	0.44
x9	0.32	0.1	0.2	···	···	0.06	0.66	0
x10	0.24	0	0.02	···	···	0.6	0.03	0.33
x11	0.12	0.45	0.44	···	···	0.64	0.45	1
x12	0.36	0.58	0.12	···	···	0.73	0.58	0.67
x13	0.2	0.02	0.24	···	···	0.34	0.02	0.89
x14	0.24	0.92	0.33	···	···	0.24	0.93	0.56

The objective function: the mean square error of prediction by X

noise

- This will introduce noise and may deteriorate the solution quality.

Scalable w.r.t. Data Volume

- Resampling: independently evaluate the fitness of a solution for *k* times and output the average.

- Resampling can reduce the time complexity of an EA from exponential to polynomial.

Theorem 1. *For the (1+1)-EA solving the OneMax problem under one-bit noise with* $p = 1$*, the expected running time is exponential.* → not use resampling

Theorem 2. *For the (1+1)-EA solving the OneMax problem under one-bit noise with* $p = 1$*, if using sampling with* $k = 2$*, the expected running time is exponential.*

Theorem 3. *For the (1+1)-EA solving the OneMax problem under one-bit noise with* $p = 1$*, if using sampling with* $k = 3$*, the expected running time is* $O(n \log n)$*.*

The sample size should be carefully selected

use resampling

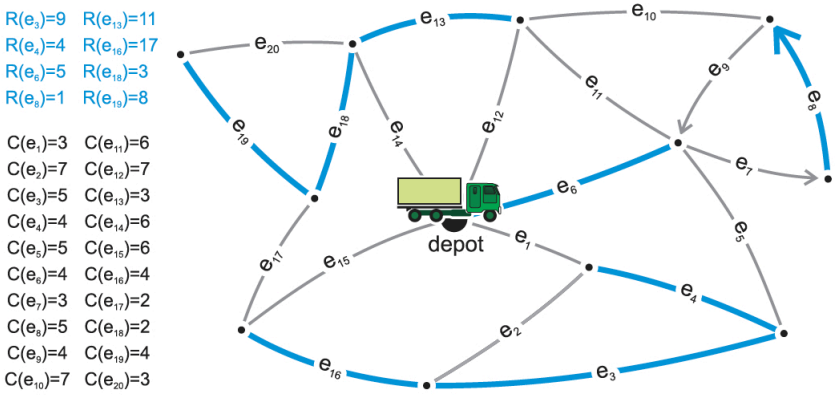
C. Qian, Y. Yu, K. Tang, Y. Jin, X. Yao and Z.-H. Zhou, "On the Effectiveness of Sampling for Evolutionary Optimization in Noisy Environments," ***Evolutionary Computation***, 26(2): 237-267, June 2018.

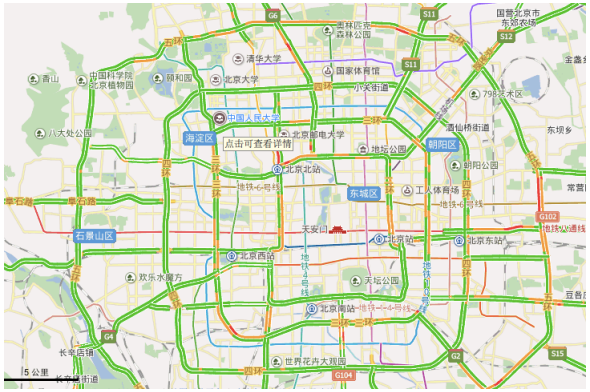
Outline

- Introduction
- General Ideas and Methodologies
- **Case Studies**
- Summary and Discussion

Case Study (1)

- **SAHiD** for Capacitated Arc Routing Problem





- Beijing: more than **3500** roads/edges (within 5-ring).
- Hefei: more than **1200** roads/edges
- Only **less than 400** roads are considered in existing benchmark.
- An almost real-world case from JD: solving a CARP with 1600 edges for every 5 minutes (emerged with the availability of big data).

Case Study (1)

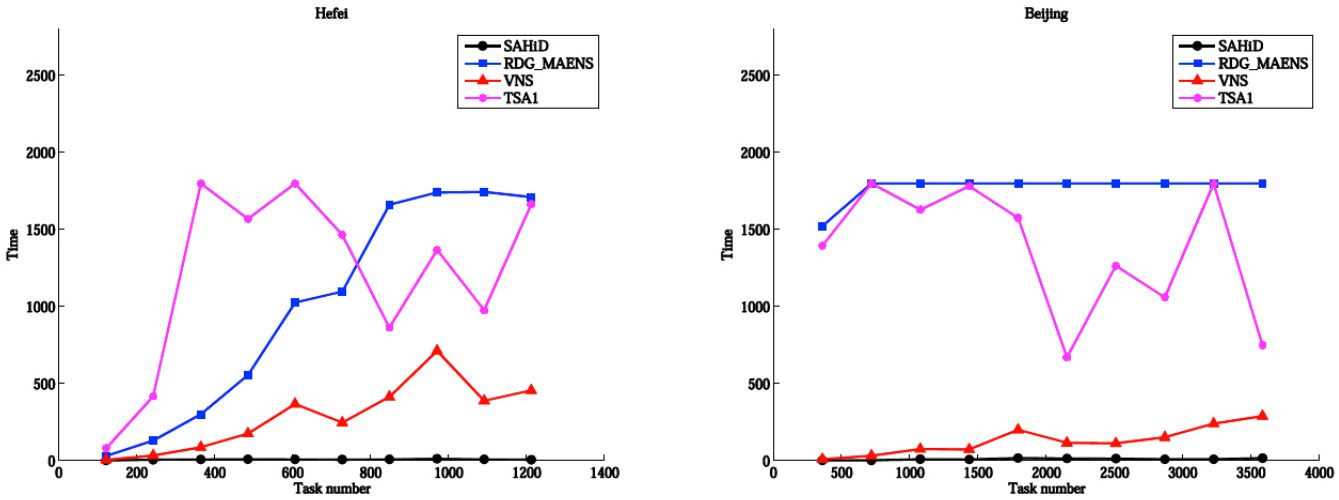
- Qualities of the solutions obtained using 30 minutes.

Name	$\|V\|$	$\|T\|$	$\|E\|$	Q	SAHiD			RDG-MAENS			VNS			TSA1		
					Best	Average	Std	Best	Average	Std	Best	Average	Std	Best	Average	Std
Beijing-1	2820	3584	358	25000	775523	784727	5591	812647	829406	12688	774502	782415*	4452	813907	829132	6340
Beijing-2	2820	3584	717	25000	1167480	1183955*	8431	1303570	1337954	18939	1168190	1192292	10196	1353567	1401363	25378
Beijing-3	2820	3584	1075	25000	1586180	1605846*	9231	1777852	1847922	33258	1591540	1618484	11888	1678224	1709279	14801
Beijing-4	2820	3584	1434	25000	1910880	1936994*	11694	2126151	2193399	34159	1920330	1953892	16746	2053938	2070885	14532
Beijing-5	2820	3584	1792	25000	2273080	2298630*	16879	2581910	2639458	32481	2293120	2335915	23040	2396483	2440319	26726
Beijing-6	2820	3584	2151	25000	2664510	2707500*	18433	2968102	3047295	41112	2705060	2743677	18024	2774161	2814735	22018
Beijing-7	2820	3584	2509	25000	3013590	3038157*	15658	3331900	3388263	26081	3015790	3063813	25226	3147294	3186240	22426
Beijing-8	2820	3584	2868	25000	3283530	3313590*	21925	3584696	3697025	44951	3323850	3366215	24686	3415275	3456037	22381
Beijing-9	2820	3584	3226	25000	3621490	3684250*	32404	3934270	4061793	49504	3653630	3723830	45148	3890129	3943883	37089
Beijing-10	2820	3584	3584	25000	3935540	4004310*	29488	4206005	4353966	51063	4002040	4040694	27384	4066188	4103532	15501
# of "w-d-l"									10-0-0			9-1-0			10-0-0	

- SAHiD is better than any other methods on 9/10 instances, except one lose on a relatively small case.

Case Study (1)

- Runtime for the state-of-the-arts to achieve the same solution quality as achieved by SAHiD in 30 seconds.



- Solution found by SAHiD in **30 seconds** can be better than those found by other methods in **30 minutes**.

K. Tang, J. Wang X. Li and X. Yao, "A Scalable Approach to Capacitated Arc Routing Problems Based on Hierarchical Decomposition," ***IEEE Transactions on Cybernetics***, 47(11): 3928-3940, November 2017.

Case Study (2)

Subset selection is to select a subset of size B from a total set of n items for optimizing some objective function

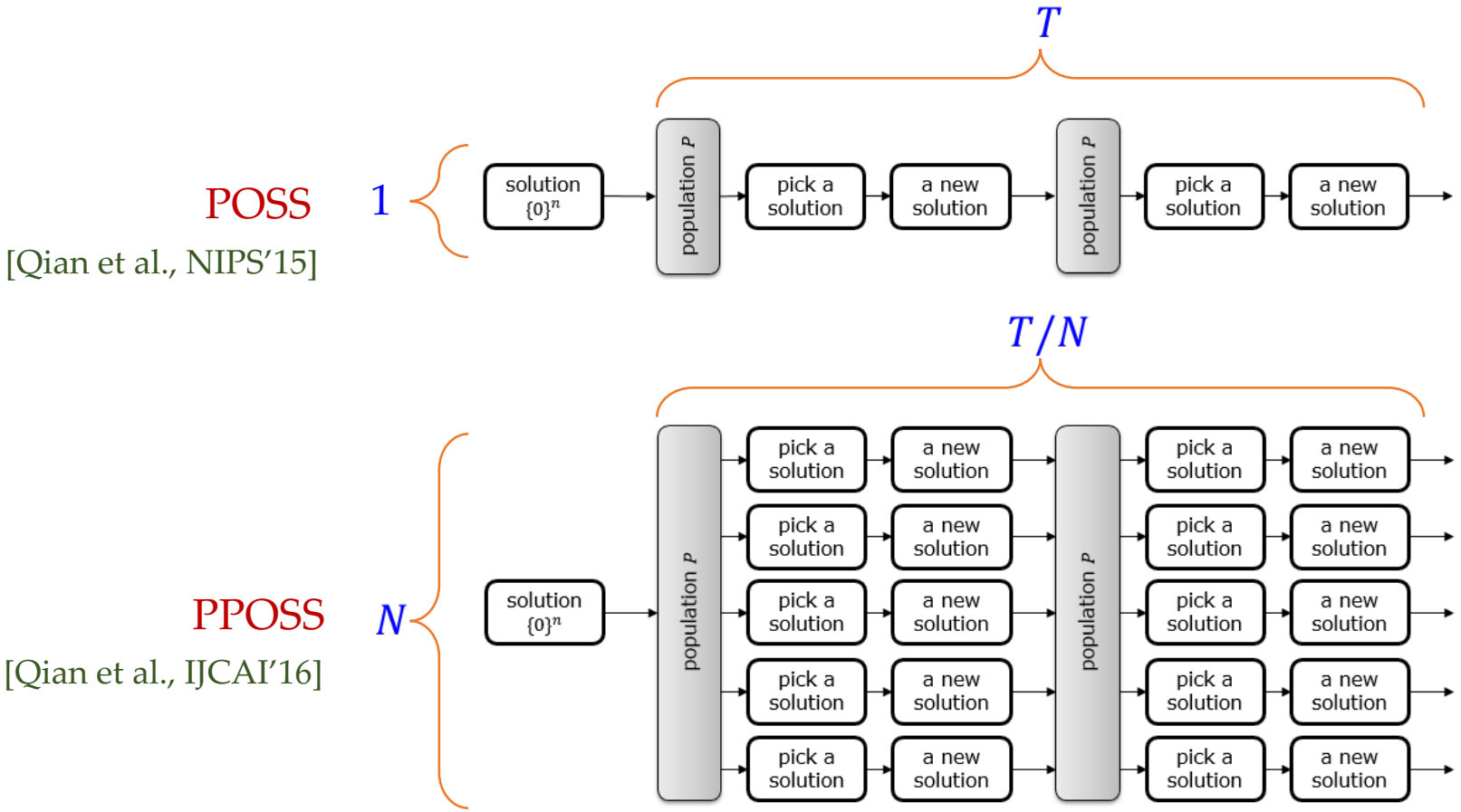
Formally stated: given all items $V = \{v_1, \ldots, v_n\}$, an objective function $f: 2^V \rightarrow \mathrm{R}$ and a budget B, it is to find a subset $X \subseteq V$ such that

$$max_{X \subseteq V} \quad f(X) \quad s.t. \quad |X| \leq B.$$

Application	v_i	f
maximum coverage	a set of elements	size of the union
sparse regression	an observation variable	MSE of prediction
influence maximization	a social network user	influence spread
document summarization	a sentence	summary quality
sensor placement	a place to install a sensor	entropy

Many applications, but NP-hard in general!

Case Study (2)



Q: the same solution quality? Yes!

C. Qian, J.-C. Shi, Y. Yu, K. Tang, and Z.-H. Zhou. Parallel Pareto Optimization for Subset Selection. In: Proceedings of ***IJCAI'16***, New York, NY, 2016, pp.1939-1945

Case Study (2)

Theorem 1. For maximizing a monotone function under the set size constraint, the expected number of iterations until PPOSS finds a solution s with $|s| \leq k$ and $f(s) \geq (1 - e^{-\gamma_{\min}}) \cdot OPT$, where $\gamma_{\min} = \min_{s:|s|=k-1} \gamma_{s,k}$, is

(1) if $N = o(n)$, then $\mathbb{E}[T] \leq 2ek^2n/N$;

(2) if $N = \Omega(n^i)$ for $1 \leq i \leq k$, then $\mathbb{E}[T] = O(k^2/i)$;

(3) if $N = \Omega(n^{\min\{3k-1,n\}})$, then $\mathbb{E}[T] = O(1)$.

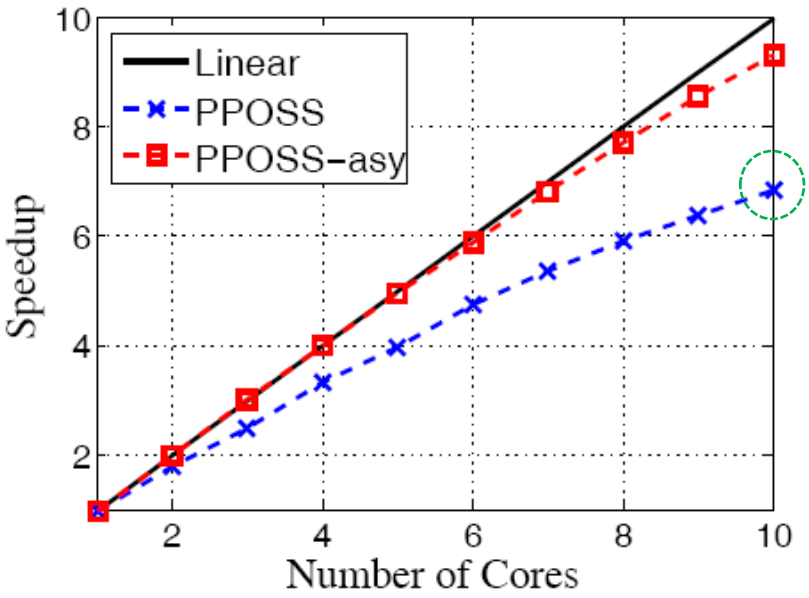
achieve the best known performance guarantee

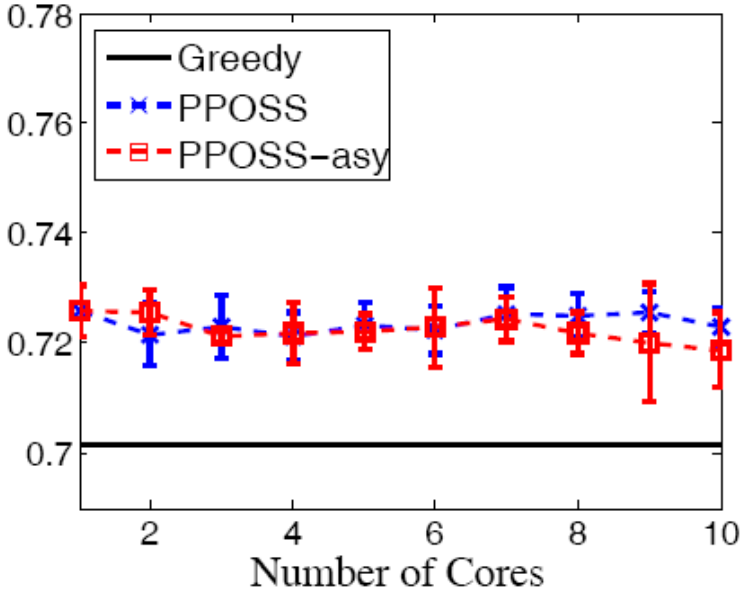
Good parallelization properties:

- When the number of processors is limited, the number of iterations can be reduced linearly w.r.t. the number of processors.
- With increasing number of processors, the number of iterations can be continuously reduced, eventually to a constant.

Case Study (2)

speedup as well as the solution quality with different number of cores





PPOSS (blue line): achieve speedup around 7 when the number of cores is 10; the solution qualities are stable

PPOSS-asy (red line): achieve better speedup (avoid the synchronous cost)**; the solution qualities are slightly worse** (the noise from asynchronization)

Case Study (3)

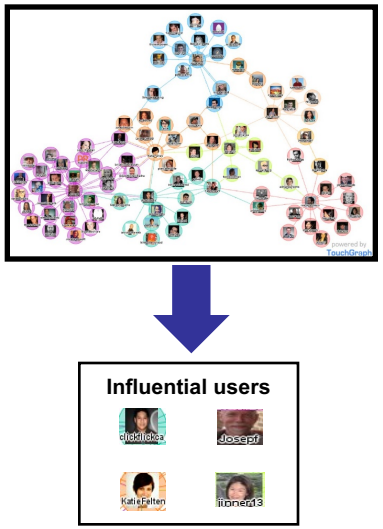
Influence maximization: select a subset of users from a social network to maximize its influence spread.

Formally stated: given a directed graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$, edge probabilities $p_{u,v}$ $((u, v) \in E)$ and a budget k, it is to find a subset $X \subseteq V$ such that

$$max_{X \subseteq V} \quad f(X) = \sum_{i=1}^{n} p(X \rightarrow v_i) \quad s.t. \quad |X| \leq k.$$

$f(X)$ → estimated by Monte Carlo simulations

Noise



multiplicative noise

$$(1 - \epsilon) f(x) \leq F(x) \leq (1 + \epsilon) f(x)$$

- Need polynomial-time approximation algorithm.
- Existing methods could be significantly affected by noise.

Case Study (3)

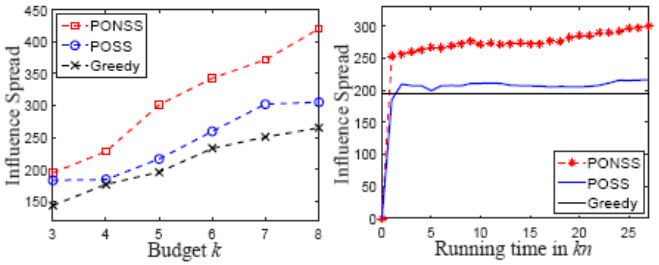
- PONSS: Pareto Optimization for Noisy Subset Selection
  - Transform the SS problem to a bi-objective optimization problem.
  - Introduce conservative domination to handle noise.

Approximation Guarantee (in polynomial time): $\gamma = 1$ (submodular), ϵ is a constant

PONSS $$f(X) \geq \frac{1-\epsilon}{1+\epsilon}\left(1-\left(1-\frac{\gamma}{k}\right)^k\right) \cdot OPT$$ constant

IV significantly better

Greedy $$f(X) \geq \frac{1}{1+\frac{2\epsilon k}{(1-\epsilon)\gamma}}\left(1-\left(\frac{1-\epsilon}{1+\epsilon}\right)^k\left(1-\frac{\gamma}{k}\right)^k\right) \cdot OPT$$ $\Theta(1/k)$



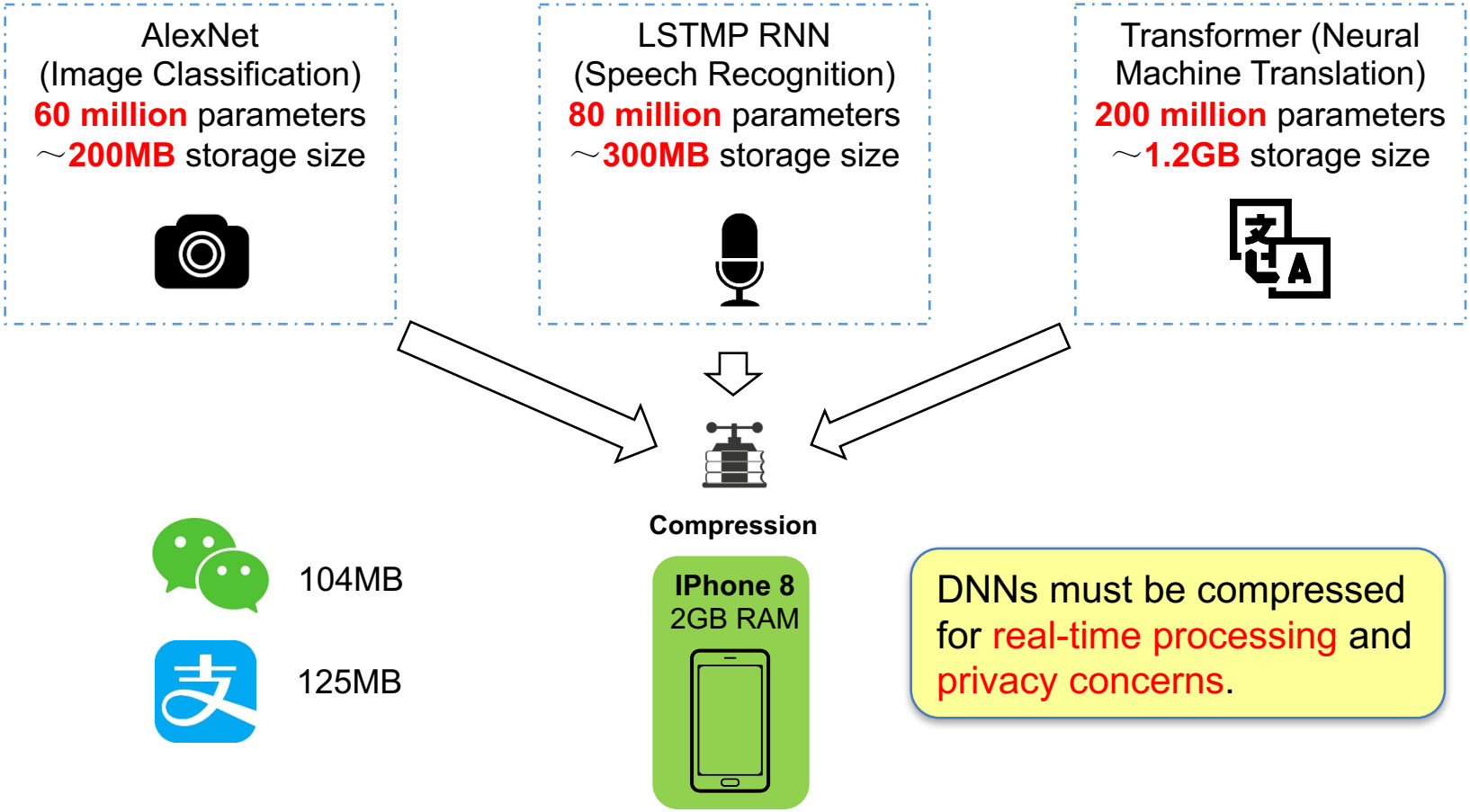
(b) *Weibo* (10,000 #nodes, 162,371 #edges)

Significantly better bound has also been proved for additive noise.

C. Qian, J. Shi, Y. Yu, K. Tang and Z.-H. Zhou, "Subset Selection under Noise," In ***NIPS'17***.

Case Study (4)

- Deep Neural Networks (DNNs) is not cost-effective, i.e., suffer from considerable redundancy and prohibitively large for mobile devices.

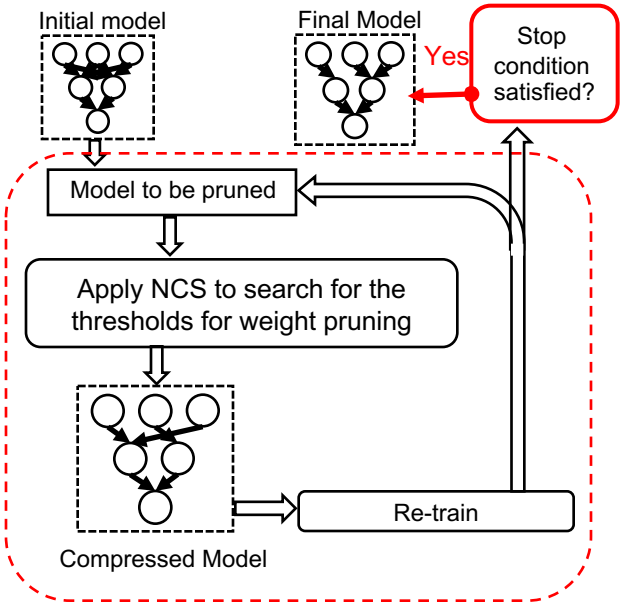


Case Study (4)

- Pruning is a typical approach for compressing DNNs.
- DNN pruning is a constrained multimodal optimization problem.

$$W^* = \underset{W' \subseteq W}{arg\min} |W'| \; s.t. \; f(W) - f(W') \leq \delta$$

– OLMP: Employs NCS as a key component of the pruning algorithm.



Iterative pruning

Model	Original Size	Pruning Method	Size after pruning	Accuracy Change (%)
LeNet-300-100	1.1MB	ITR, 2015	93.9KB	+0.05
		DS, 2016	20.1KB	+0.29
		SWS, 2017	49.0KB	-0.05
		Sparse VD, 2017	16.6KB	-0.28
		OLMP, 2018	**10.0KB**	+0.1
LeNet-5	3.3MB	ITR, 2015	281.6KB	+0.03
		DS, 2016	31.3KB	0
		SWS, 2017	16.9KB	-0.09
		Sparse VD, 2017	12KB	+0.05
		OLMP, 2018	**11KB**	0
AlexNet	228.0MB	**OLMP, 2018**	**2.8MB**	+0.4

G. Li, C. Qian, C. Jiang, X. Lu and K. Tang, "Optimization based Layer-wise Magnitude-based Pruning for DNN Compression," in Proc. of ***IJCAI'18***.

Outline

- Introduction
- General Ideas and Methodologies
- Case Studies
- **Summary and Discussion**

Summary and Discussion

- Scalability of evolutionary search involves many factors.

- Different factors induce different demanding issues, we considered a few cases including
  - No. of decision variables – huge search space
  - No. of processors – performance guarantee
  - Volume of data – costly fitness evaluations

- What if we have more...
  - Objective functions
  - Constraints
  - Problem instances

南方科技大学

2

计算机科学与技术

人工智能

演化计算，机器学习，认知机器人、计算智能、群体智能、智能控制、智能感知、智能无人系统等

在职：

姚新讲席教授

史玉回讲席教授

Hisao Ishibuch讲席教授

唐珂教授

郝祁副教授

程然助理教授

即将入职：

王立新教授

Tom Ko助理教授

数据科学

数据科学基础理论、数据技术及应用、大数据分析、云计算等

在职：

杨双华讲席教授

骆宗伟副教授

Adam Ghandar助理教授

唐博助理教授

即将入职：

宋轩副教授

计算机系统与网络

下一代无线网络技术，传感器网络与物联网技术，信息物理融合系统，网络控制，计算机体系结构等。

在职：

杨双华讲席教授

G.Theodoropoulos讲席教授

Elvis Sze-Yeung Liu助理教授

张进助理教授

刘烨庞助理教授

Shin Hwei Tan助理教授

张煜群访问助理教授

吴冀衍助理教授

Alia Asheralieva助理教授

计算机视觉与图形学

计算机视觉、计算机图形学、图像/视频编码传输，以及图像/视频处理等。

在职：

郝祁副教授

Luca Rossi助理教授

即将入职：

刘江教授

郑锋助理教授

计算机理论

密码学、纠错编码、算法分析和计算复杂度理论等。

在职：

姚新讲席教授

王琦助理教授

- 2016年8月正式运营
- 目前全职教师25人
- 平均每年级本科生100余人
- 2017年5月以来：
  - 广东省普通高校演化智能系统实验室
  - 深圳市计算智能重点实验室
  - JCR一区、中国计算机学会A类论文64篇

Thanks you!

Questions/comments?