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@sustech.edu.cn

SAINT Workshop on Evolutionary Computation and Its Applications, SUSTech, March 2019

Outline

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

What is Scalability?

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

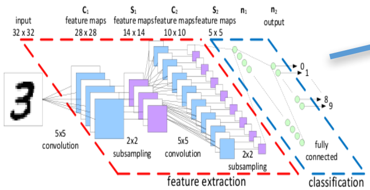
- Environmental factors
  - Decision variables
  - Data
  - Computing facilities, e.g., CPUs
  - etc.

Evolutionary Algorithms

- 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

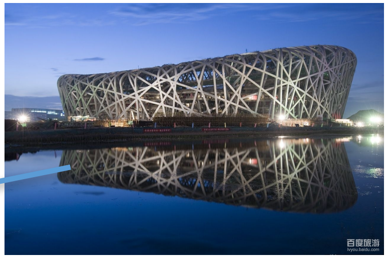
Railway timetabling



Network Optimization

Non-differentiable constraints + mixed integer search space

Non-differentiable Objective function



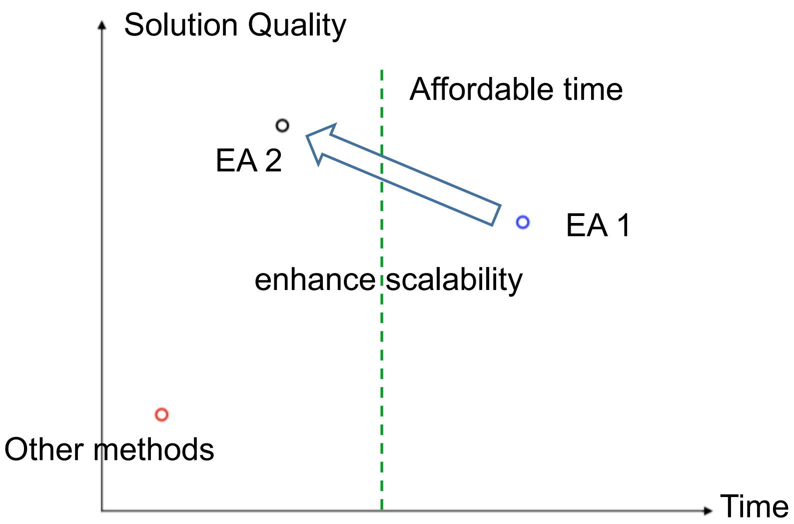
Truss design

Portfolio optimization

Discrete Search space

Why scalability for EAs?

- Theoretically, scalability plays a central role in computer science.
- Practically, to enable EAs find **good** solutions can be obtained **efficiently** (i.e., with affordable time), which depends on
  - No. of iterations/generations (relevant to No. of Decision variables)
  - Time needed for a single fitness evaluation (relevant to volume of data)
  - Hardware/infrastructure



Outline

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

Scalable w.r.t. Decision Variables

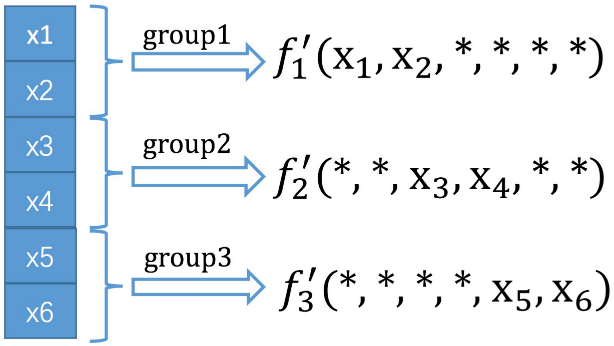
Suppose we have an optimization problem:

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

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

Scalable w.r.t. Decision Variables

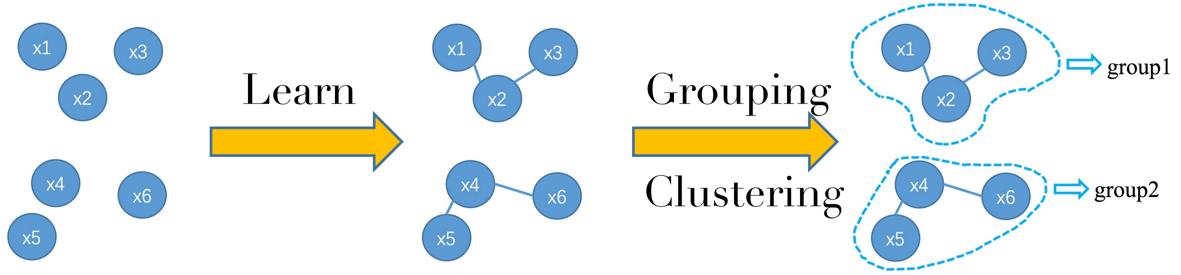
- Basic idea: Divide-and-Conquer (search in multiple subspaces)



- **Challenge**: little prior knowledge about
  - whether the objective function is separable
  - how to divide decision variables

Scalable w.r.t. Decision Variables

- **Divide**: Randomly or Learn to Group

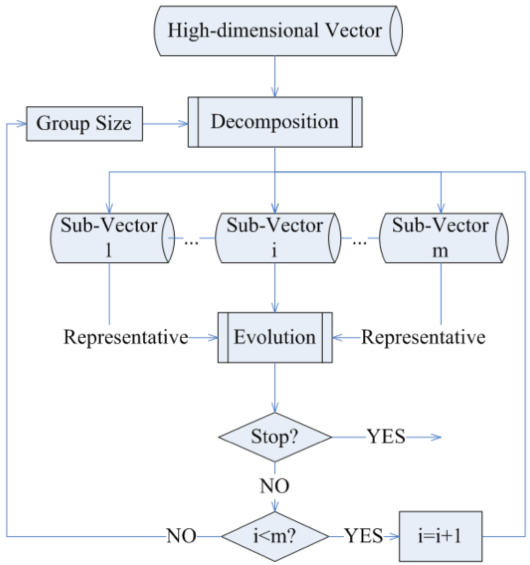


- **Conquer**: Correlate the solving phase of sub-problems, since
  - The learned relationships between variables are seldom perfect.
  - Sometimes the problem itself is not separable at all.

Z. Yang, K. Tang and X. Yao, "Large Scale Evolutionary Optimization Using Cooperative Coevolution," *Information Sciences,* 178(15): 2985-2999, August 2008. *Google Scholar Citation: 600+*

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



A natural implementation:

Cooperative Coevolution

Comparison between DECC-G and SaNSDE on functions f1 − f7 (unimodal), with dimension D = 1000, averaged over 25 runs.

	# of Dim	SaNSDE	DECC-G
f1	1000	6.97E+00	**2.17E-25**
f2	1000	1.24E+00	**5.37E-14**
f3	1000	6.43E+01	**3.71E-23**
f4	1000	4.99E+01	**1.01E-01**
f5	1000	3.31E+03	**9.87E+02**
f6	1000	3.93E+03	**0.00E+00**
*f*7	1000	1.18E+01	**8.40E-03**

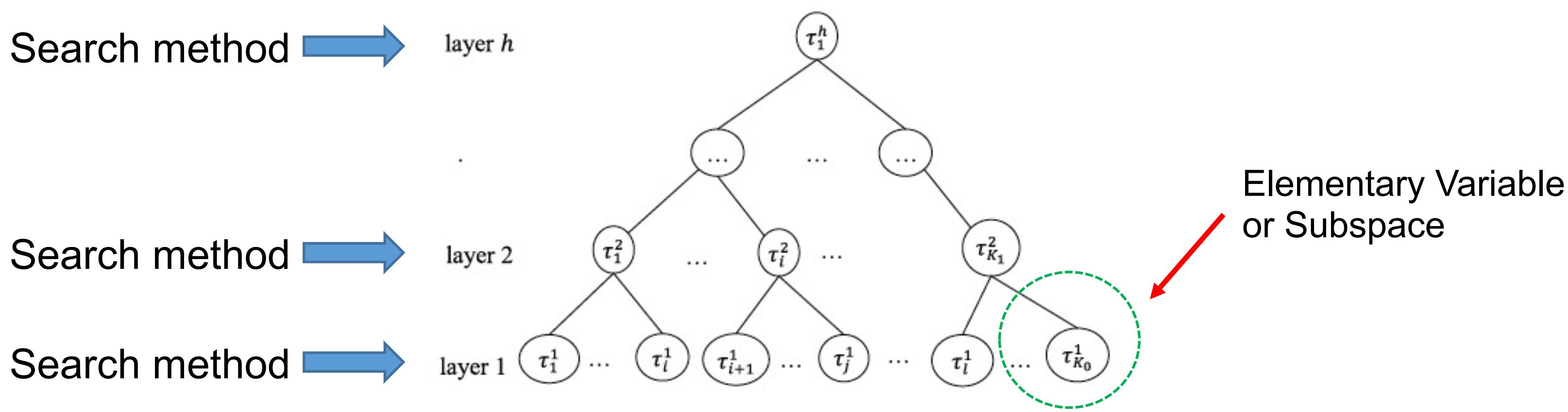
Comparison between DECC-G and SaNSDE on functions f8 − f13 (multimodal), with dimension D = 1000, averaged over 25 runs.

	# of Dim	SaNSDE	DECC-G
*f*8	1000	-372991	**-418983**
*f*9	1000	8.69E+02	**3.55E-16**
*f*10	1000	1.12E+01	**2.22E-13**
*f*11	1000	4.80E-01	**1.01E-15**
*f*12	1000	8.97E+00	**6.89E-25**
*f*13	1000	7.41E+02	**2.55E-21**

Scalable w.r.t. Decision Variables

- Dividing a problem in a "linear" way, e.g., divide *D* variables into *K* groups of size *D*/*K*, is not ideal due to the conflict between *K* and *D*/*K*.

- 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"



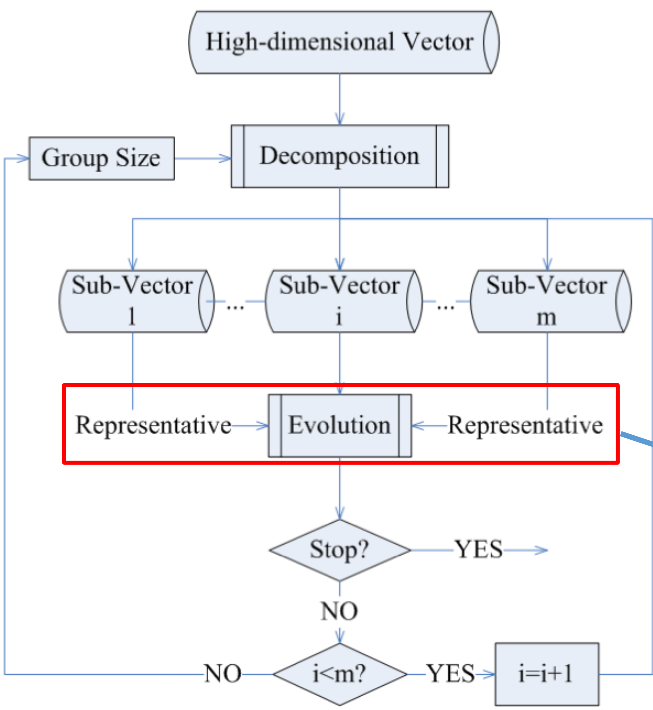
K. Tang, J. Wang X. Li and X. Yao, "A Scalable Approach to Capacitated Arc Routing Problems Based on Hierarchical Decomposition," ***IEEE Transactions on Cybernetics***, 47(11): 3928-3940, November 2017.

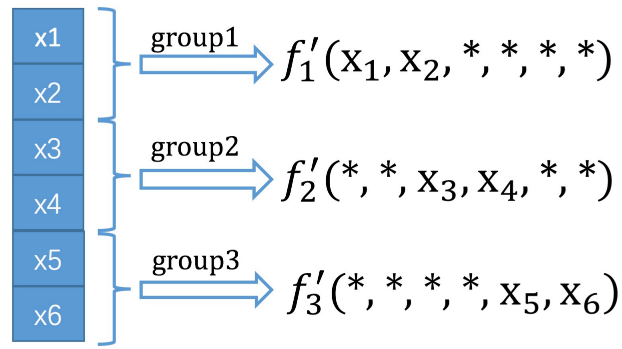
Scalable w.r.t. Processors

How fast can we solve a problem 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.

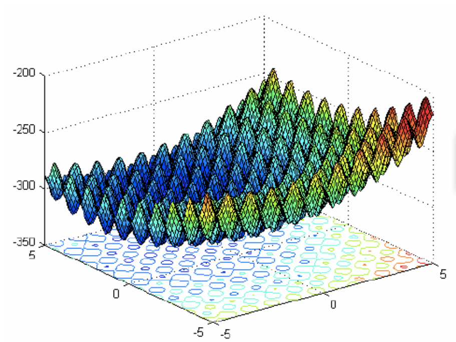




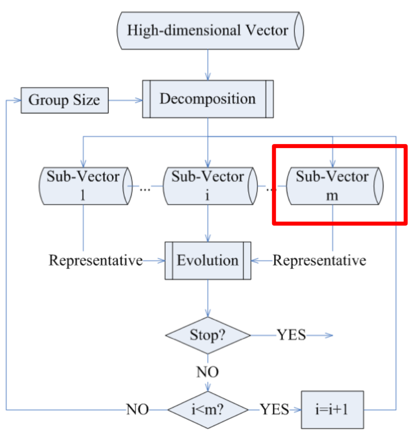
- Share all possibly good values of $x_3 \sim x_6$ with group 1.
- Expensive both in terms of communication and computation

Scalable w.r.t. Processors

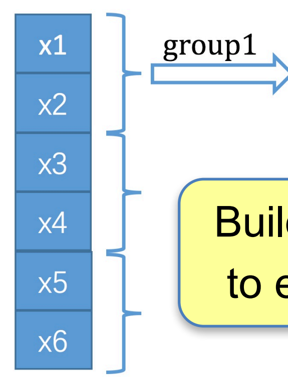
Idea: using data generated during search course to build surrogate models



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



One model for each subpopulation

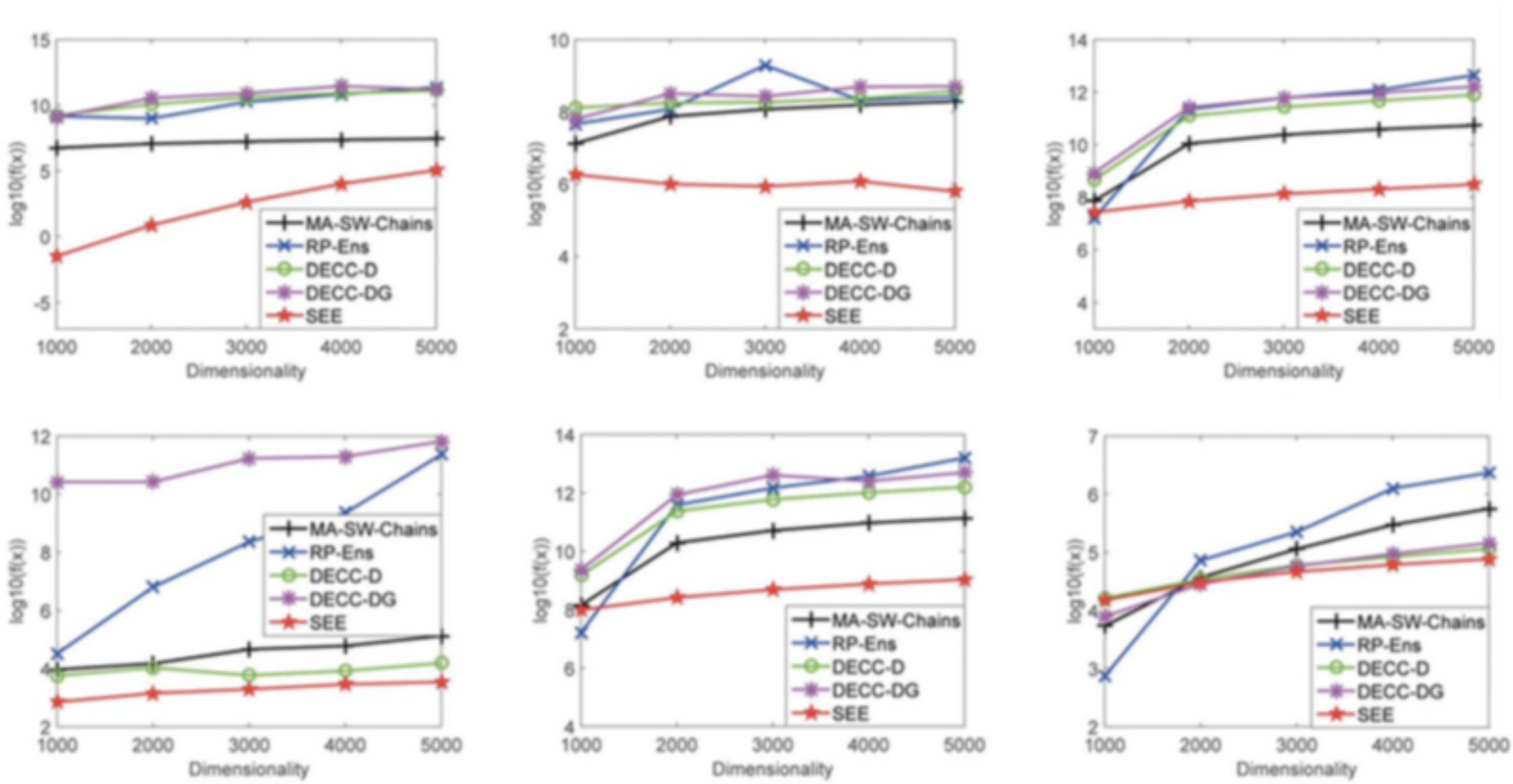


$f_1'(x_1, x_2, \underbrace{*, *, *, *})$

Build surrogate model to evaluate x_1 and x_2

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.

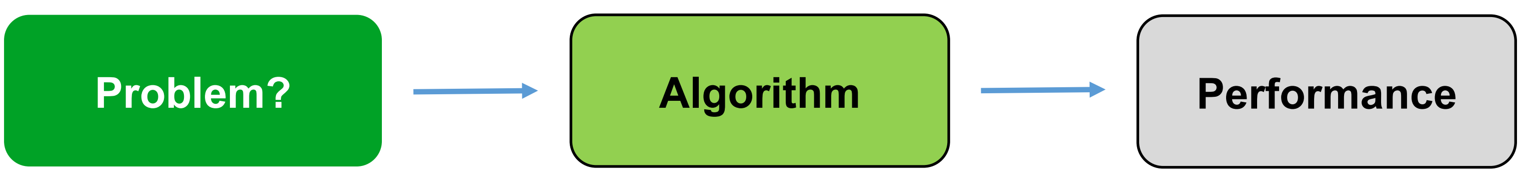
Scalable w.r.t. Processors



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.

Outline

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



Case Study (1)

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 $f: 2^V \to \mathbb{R}$ and a budget B, it is to find a subset $X \subseteq V$ such that

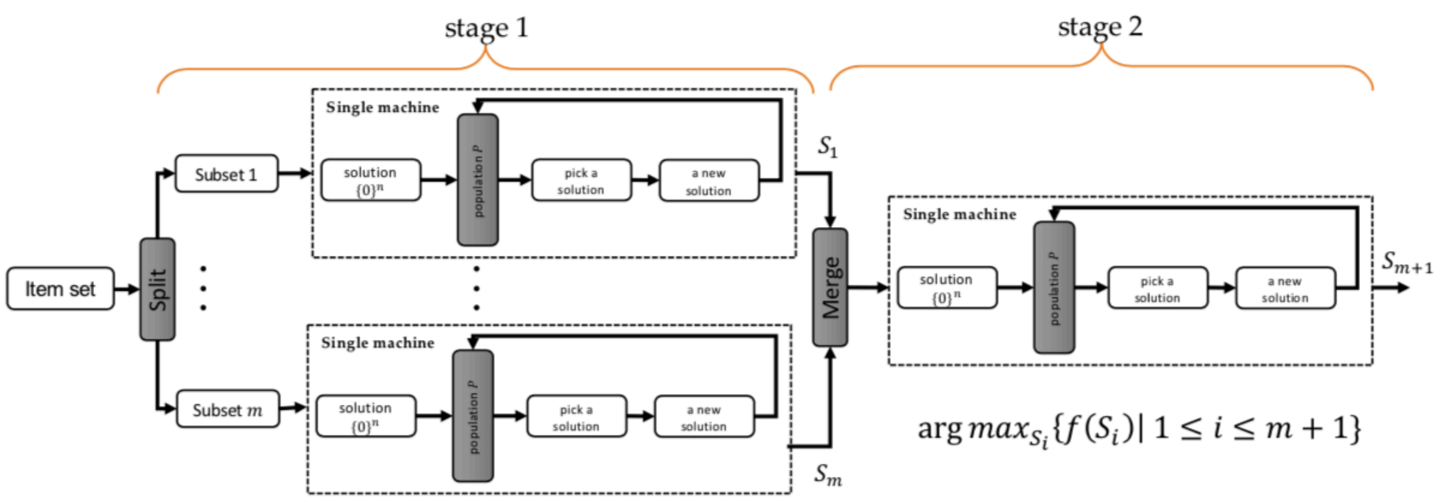
$$max_{X \subseteq V} \quad f(X) \quad s.t. \quad |X| \leq B.$$

Application	$\boldsymbol{v_i}$	$\boldsymbol{f}$
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 (1)

Divide and Conquer for large-scale subset selection



almost linear speed-up with **guaranteed Solution Quality**

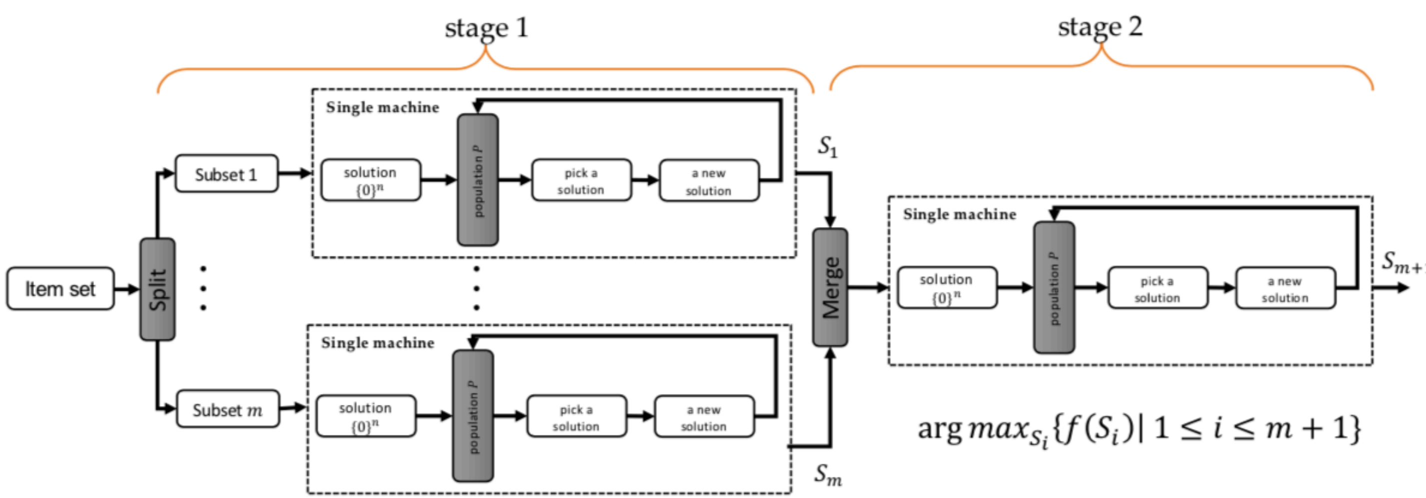
$$O\left(\frac{2eB^2n}{m}\right)$$

$$O\big(2eB^2(mk)\big)$$

C. Qian, G. Li, C. Feng and K. Tang, "Distributed Pareto Optimization for Subset Selection," In ***IJCAI'18***.

Case Study (1)

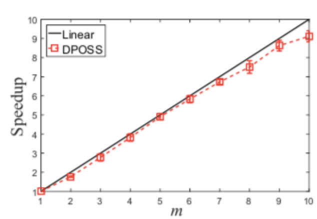
Divide and Conquer for large-scale subset selection?

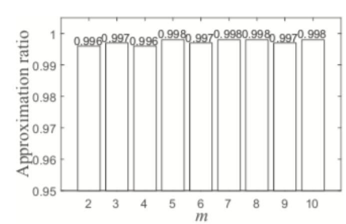


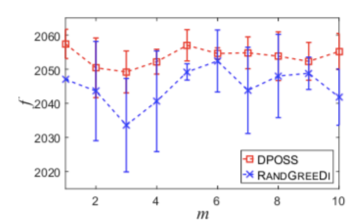
almost linear speed-up with **guaranteed Solution Quality**

$$O\left(\frac{2eB^2n}{m}\right) \qquad O\left(2eB^2(mk)\right)$$

C. Qian, G. Li, C. Feng and K. Tang, "Distributed Pareto Optimization for Subset Selection," In ***IJCAI'18***.

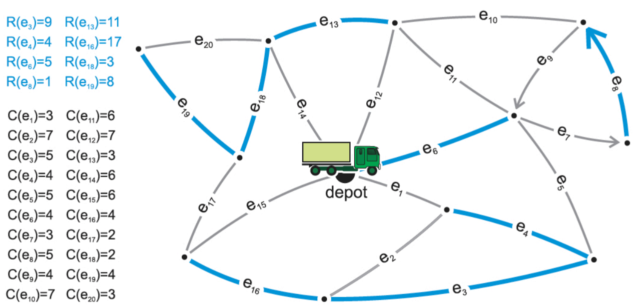


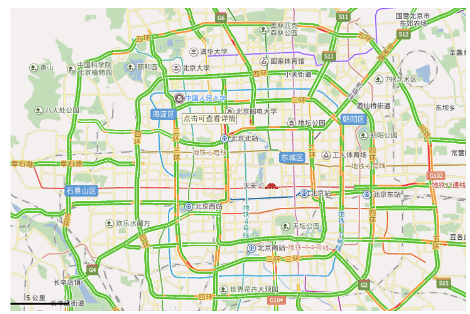


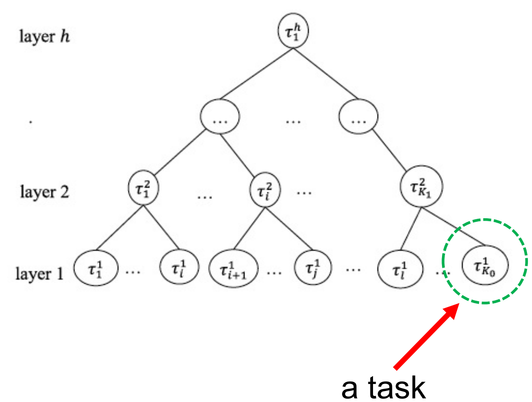


Case Study (2)

- **SAHiD** for Capacitated Arc/Vehicle Routing Problem (CARP)
- Idea: Hierarchical Decomposition + Local Search







Research:

- Benchmarks involve less than **400** roads/edges

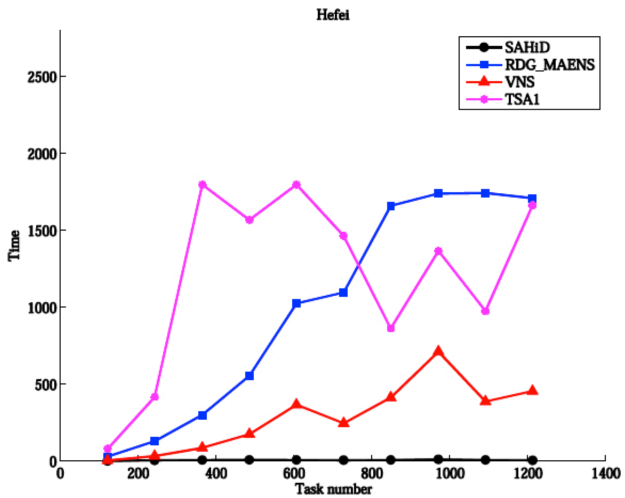
Reality:

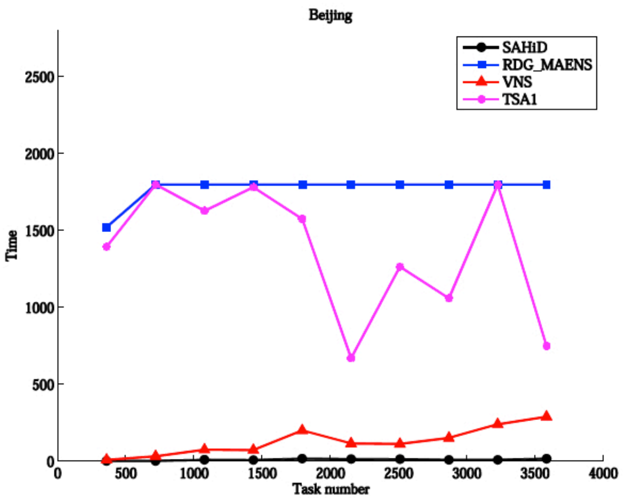
- Beijing: more than **3500** roads (within 5-ring)
- JD: solving CARP with **1600 edges in 5 minutes**

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)

- 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**.

Case Study (2)

- Qualities of the solutions obtained using 30 minutes.

Name	$\lvert V\rvert$	$\lvert T\rvert$	$\lvert E\rvert$	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 (3)

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

AlexNet
(Image Classification)
60 million parameters
～**200MB** storage size

LSTMP RNN
(Speech Recognition)
80 million parameters
～**300MB** storage size

Transformer (Neural Machine Translation)
200 million parameters
～**1.2GB** storage size

Compression

104MB

125MB

IPhone 8
2GB RAM

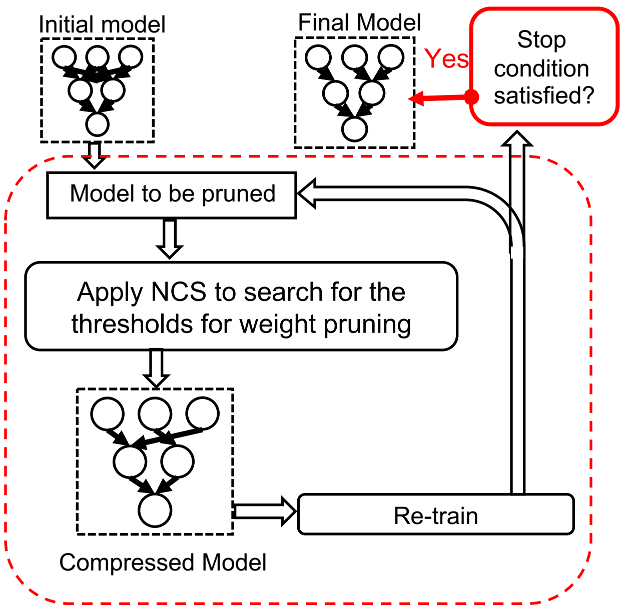
DNNs must be compressed for real-time processing and privacy concerns.

Case Study (3)

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

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

– 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 Methodologies
- Case Studies
- **Summary and Discussion**

Summary

Motivation	Making EAs faster (while keeping the solution quality)		
Factors	No. of Decision Variables Huge Search Space	No. of Processors Practical Efficiency	
Case Studies	Subset Selection	Vehicle Routing	DNN Compression

Discussions

Other interesting topics/research questions on “scalability” + “EA”

- **Any new challenge brought by Large-Scale Multi-Objective Optimization?**

1. 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*, in press

- **Larger population (more processors) leads to less iterations?**

2. T. Chen, K. Tang, G. Chen and X. Yao, “A Large Population Size Can Be Unhelpful in Evolutionary Algorithms,” *Theoretical Computer Science*, 436: 54-70, June 2012.
3. C. Qian, J.-C. Shi, Y. Yu, K. Tang and Z.-H. Zhou, “Parallel Pareto Optimization for Subset Selection”, in *IJCAI-2016*

- **Parallel search methods (solvers) for reliable optimization system?**

4. F. Peng, K. Tang, G. Chen and X. Yao, “Population-based Algorithm Portfolios for Numerical Optimization,” IEEE Transactions on Evolutionary Computation, 14(5): 782-800, October 2010.
5. S. Liu, K. Tang and X. Yao, “Automatic Construction of Parallel Portfolios via Explicit Instance Grouping,” in *AAAI- 2019*.

- **Data intensive fitness evaluation (EA + Noisy Optimization?)**

6. C. Qian, J. Shi, Y. Yu, K. Tang and Z.-H. Zhou, “Subset Selection under Noise,” in *NIPS-2017*.

Discussions

Other interesting topics/research questions on “scalability” + “EA”

- **Any new challenge brought by Large-Scale Multi-Objective Optimization?**

1. 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*, in press

- **Larger population (more processors) leads to less iterations?**

2. T. Chen, K. Tang, G. Chen and X. Yao, “A Large Population Size Can Be Unhelpful in Evolutionary Algorithms,” *Theoretical Computer Science*, 436: 54-70, June 2012.
3. C. Qian, J.-C. Shi, Y. Yu, K. Tang and Z.-H. Zhou, “Parallel Pareto Optimization for Subset Selection”, in *IJCAI-2016*

- **Parallel search methods (solvers) for reliable optimization system?**

4. F. Peng, K. Tang, G. Chen and X. Yao, “Population-based Algorithm Portfolios for Numerical Optimization,” IEEE Transactions on Evolutionary Computation, 14(5): 782-800, October 2010.
5. S. Liu, K. Tang and X. Yao, “Automatic Construction of Parallel Portfolios via Explicit Instance Grouping,” in *AAAI- 2019*.

- **Data intensive fitness evaluation (EA + Noisy Optimization?)**

6. C. Qian, J. Shi, Y. Yu, K. Tang and Z.-H. Zhou, “Subset Selection under Noise,” in *NIPS-2017*.

Thanks you!

Questions/comments?