Challenges in applying Evolutionary Algorithms to real-world problems

Professor Xiaodong Li

School of Science (Computer Science and Software Engineering)

RMIT University, Melbourne, Australia

Email: xiaodong.li@rmit.edu.au

RMIT UNIVERSITY

Why this talk?

- Personal experience when talking to optimization researchers from math and operations research backgrounds;
- Dealing with practitioners who need solutions to their real-world problems;
- Growing popularity of nature-inspired optimization techniques;
- Clear gaps between existing methods and practical problems to be solved;
- Human-in-the-loop approach to optimization;
- Personal views on research needs and motivations.

My pathway to become an EC researcher

- Graduated from Xidian University with a bachelor in Information Science;
- Master and PhD in Artificial Intelligence, Otago University, New Zealand;
- First academic job as a lecturer at Charles Sturt University, then Monash, and finally at RMIT University;
- In early years interested in artificial life, complexity, and swarming behaviour;
- After a PhD on “"Connectionist Learning Architecture Based on an Optical Thin-Film Multilayer Model“, looked for new research ideas…
- Developed a fire-spread simulation model using cellular automata (i.e., an artificial life model), which led to my interests in swarm intelligence;
- Attended Swarm Fest in 2001, and first time at CEC in 2002.
- Attended first GECCO in 2003; my first ever GECCO paper, on a multiobjective PSO algorithm won the **ACM SIGEVO Impact Award in 2013** for the highest citations among all GECCO’03 paper.
- Visiting research fellow to Prof. Xin Yao at the University of Birmingham in 2008.

Li, X. (2003), "A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization", in Proceeding of Genetic and Evolutionary Computation Conference 2003 (GECCO'03), Lecture Notes in Computer Science (LNCS 2723),eds. Erick Cantu-Paz et al., Chicago, USA, 12-16, July, 2003, pp.37-48.

My pathway to become an EC researcher

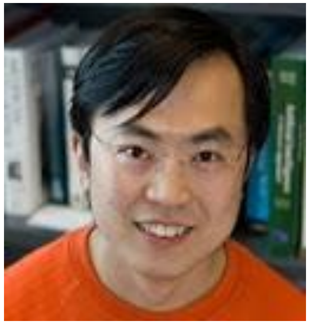
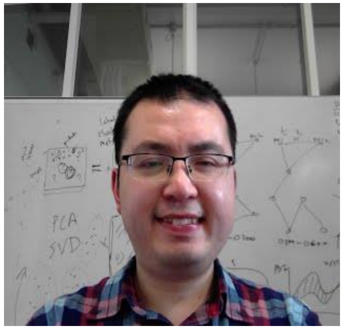
ecml
Evolutionary Computation and Machine Learning Group

CEC'17 conference, San Sebastián, Spain, receiving 2017 IEEE CIS TEVC Outstanding Paper Award.

ecml Evolutionary Computation and Machine Learning

We study and develop ***nature-inspired*** computational models and algorithms, especially in the areas of evolutionary computation and machine learning, and apply them to real-world problems. The group takes an inter-disciplinary approach drawing its inspirations from mathematical programming, meta-heuristics, and operations research.




Staff: Prof. Xiaodong Li (Group leader), A/Prof. Vic Ciesielski, Dr. Andy Song, Dr. Jeffrey Chan, A/Prof. Fabio Zambetta
Students: more than 15 PhD candidates, plus several master and honours students
Teaching: Artificial Intelligence, Machine Learning, Evolutionary Computing, and Data Mining

Further information: https://titan.csit.rmit.edu.au/~e46507/ecml/

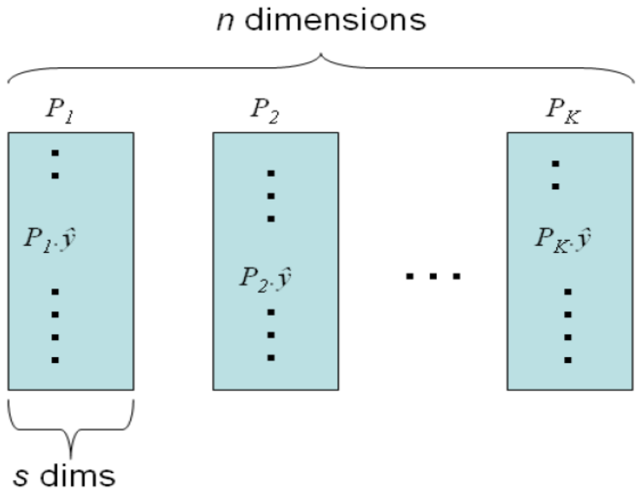
ECML members

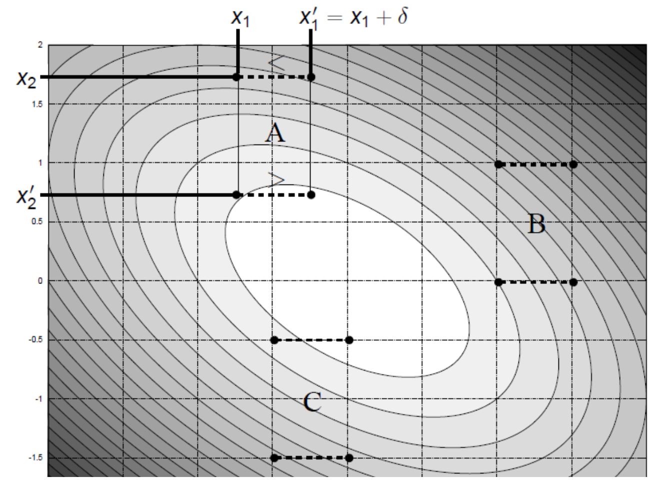
My research activities so far have been largely focussed on algorithmic development or enhancement.

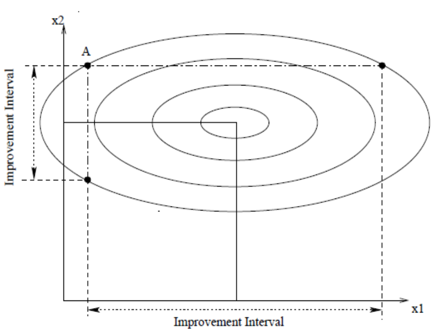
Next few slides are some examples…

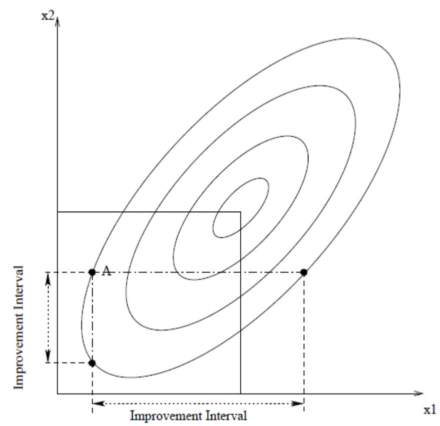
But my interests have gradually shifted towards solving more practically relevant problems.

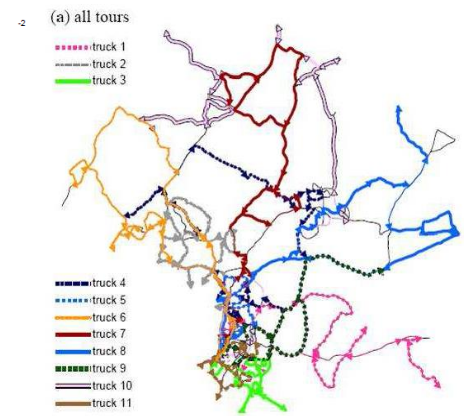
Divide-and-Conquer: Large Scale Global Optimization with variable grouping techniques











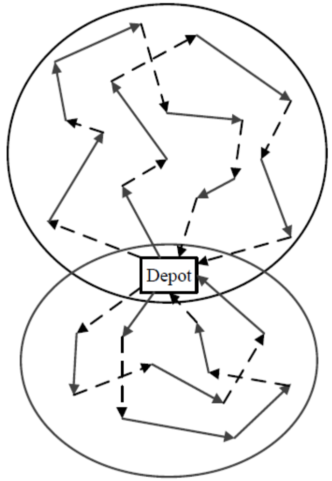
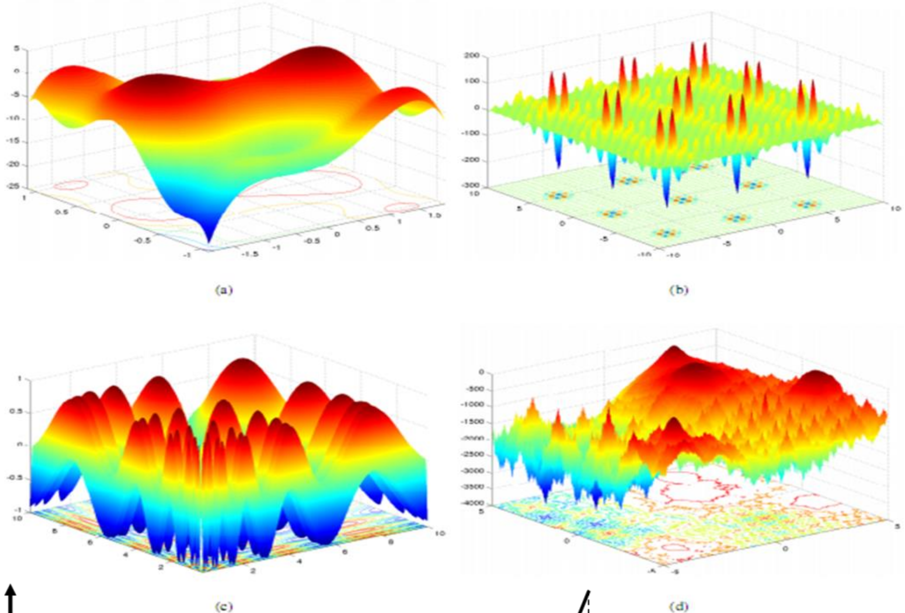
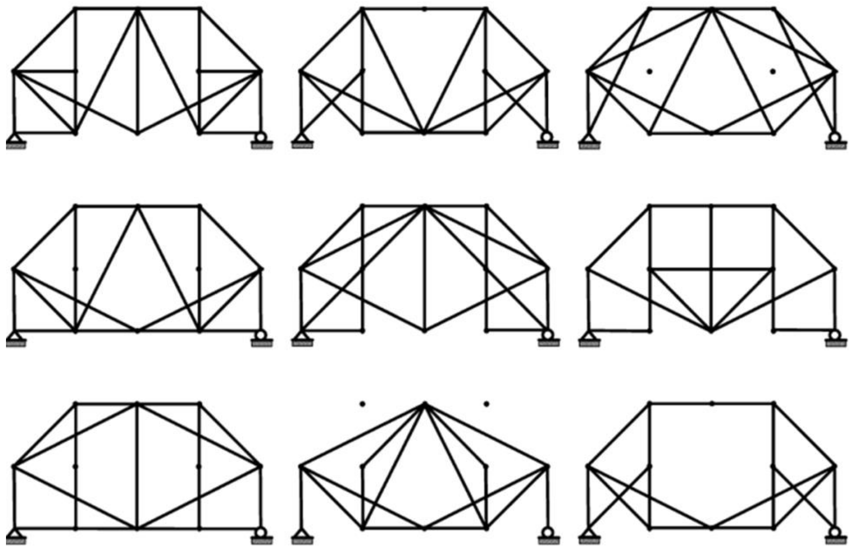


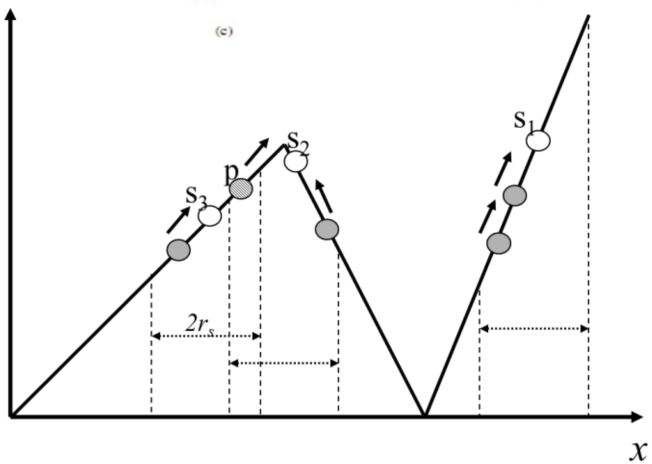
Figure: A routing plan

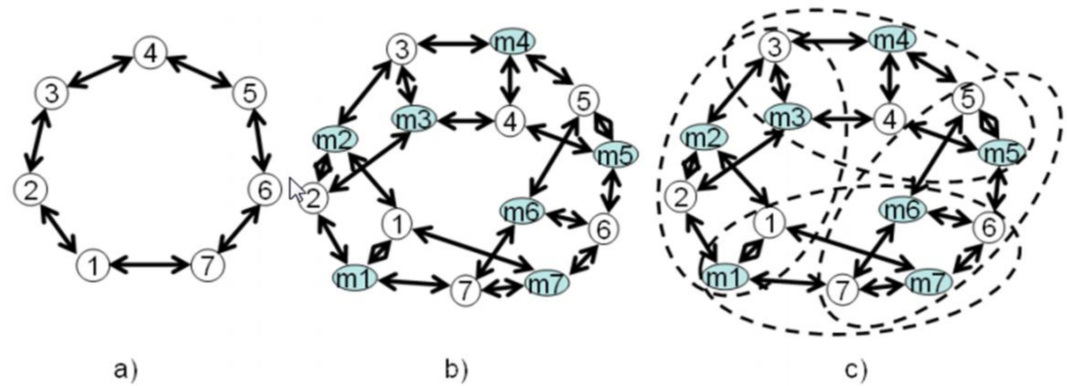
Omidvar, M.,**Li, X.** Mei, Y. Yao, X. (2014), "Cooperative Co-evolution with Differential Grouping for Large Scale Optimization", *IEEE Transactions on Evolutionary Computation,* 18(3): 378-393, June 2014 (**2017 IEEE CIS "IEEE Transactions on Evolutionary Computation Outstanding Paper Award**).
Mei, Y.,Li, X. and Yao, X. (2014), "Cooperative Co-evolution with Route Distance Grouping for Large-Scale Capacitated Arc Routing Problems", IEEE Transactions on Evolutionary Computation, 18(3): 435-449, June 2014.

Multi-modal Optimization using Niching Methods





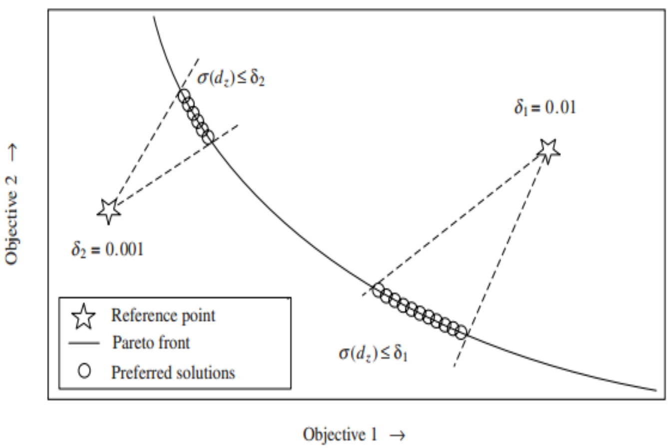


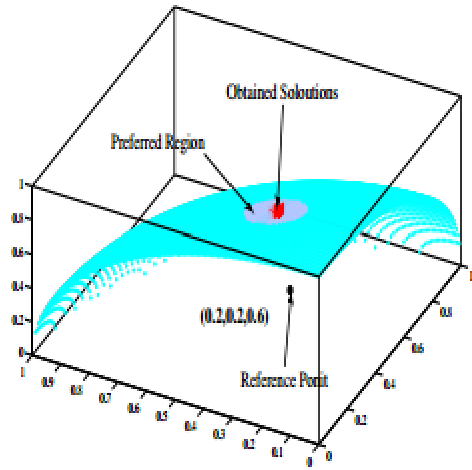


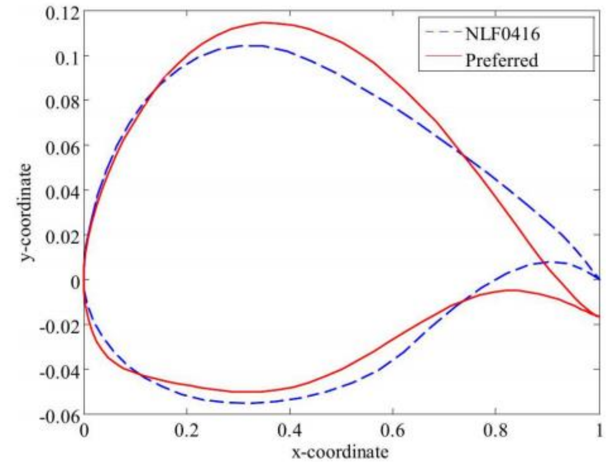
X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. on Evol. Comput.*, vol. 14, no. 1, pp. 150 – 169, February 2010.

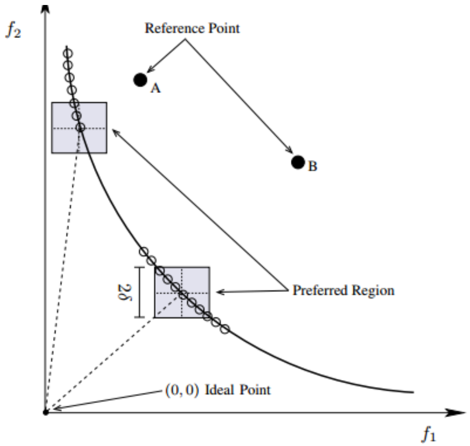
X. Li, A. Engelbrecht, and M. Epitropakis, "Benchmark functions for cec'2013 special session and competition on niching methods for multimodal function optimization," Technical Report, Evolutionary Computation and Machine Learning Group, RMIT University, 2013.

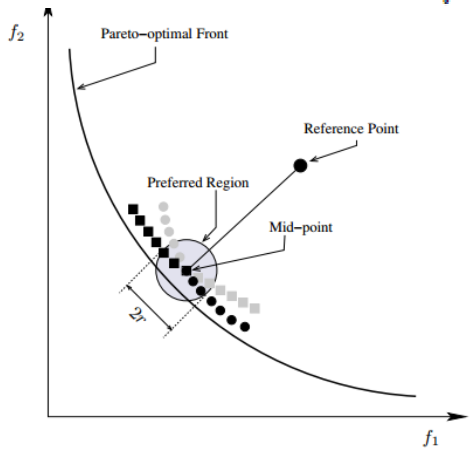
Preference-based Evolutionary Multiobjective Optimization











Mohammadi, A., Omidvar, M. and Li, X. (2013), "A New Performance Metric for User-preference Based Multi-objective Evolutionary Algorithms", in Proceedings of Congress of Evolutionary Computation (CEC 2013), IEEE, pp.2825 - 2832.
Carrese, R., Sobester, A., Winarto, H., and Li, X. (2011), "Swarm heuristic for identifying preferred solutions in surrogate-based multiobjective engineering design", American Institute of Aeronautics and Astronautics Journal, 49(7): 1437- 1449, July 2011.

Shifting towards solving more practical problems

- When interacting with operations research community;
- NICTA/Data61 optimization summer schools;
- People in engineering and mathematical programming (RMITOpt group, AMSI Optimise 2017, Data61 talk series);
- PhD projects with more practical problems;
- Personal communication with Prof. Zbigniew Michalewicz; his company SolveIT (specialised in integrated planning and supply chain optimization) winning big contracts with some of the largest companies in Australia such as BHP, Rio Tinto.

- **Changing from more algorithms focused to more problem focused**.

Michalewicz, Z., The Emperor is Naked: Evolutionary Algorithms for Real-World Applications ACM Ubiquity, November 2012, pp. 1 - 13.
Michalewicz, Z., Quo Vadis, Evolutionary Computation? On a growing gap between theory and practice, Springer LNCS State-of-the-Art Survey, J. Liu, C. Alippi, B. Bouchon-Meunier, G. Greenwood, H. Abbass (Editors), 2012.

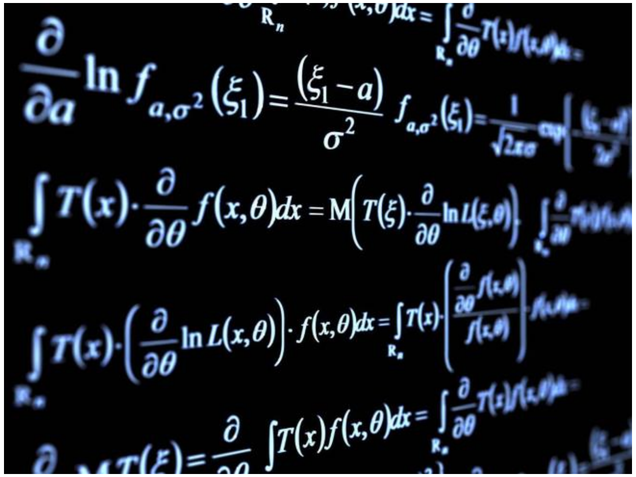
Strengths of nature-inspired optimization methods

- Pros
  - Robust and generic solution methods;
  - No gradient information is required;
  - Less demanding on rigorous math formulation;
  - Usually work with a population of candidate solutions (implicit parallelism);
  - Strong global search capability, i.e., less prone to getting stuck on local optima, and work well on non-convex problems;
  - Fewer assumptions

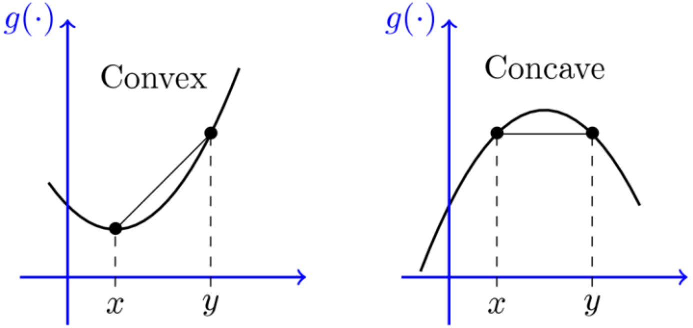
- Cons
  - Weak theoretical foundation; less established;
  - Computationally more expensive, as compared with conventional methods;
  - Little consideration on solution **constructive** approaches;
  - Difficult to analyse population dynamics;
  - Difficult to apply to complex large-scale problems with multitudes of components inter-dependent to each other.

Mathematical programming methods

- Pros
  - Strong theoretical foundation, more established;
  - Work well on linear and integer programming problems; if problems are not linear, and approximation can be still good enough for certain problems;
  - Work well on problems with convex shapes;
  - “Construct and search” approach can be effective;
  - Some effective problem reduction techniques;
  - Usually single-solution search methods.



- Cons
  - Often have strong assumptions, e.g., convexity;
  - Non-convex and nonlinear problems are much harder to deal with; unfortunately many real-world problems belong to this class;
  - MIP solvers such as CPLEX or GUROBI often can only solve small or medium sized problem instances, e.g., the branch-and-bound methods.



Mathematical programming methods

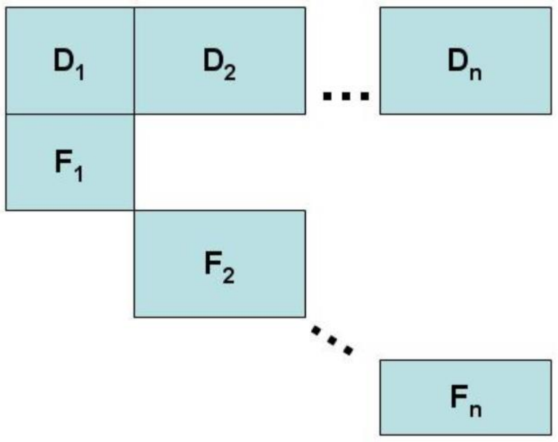
- **Lagrangian relaxation**:

$$\begin{aligned} \text{Minimize} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in X \end{aligned}$$

$$\begin{aligned} \text{Minimize} \quad & \mathbf{c}^T\mathbf{x} + \mu^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \\ \text{subject to} \quad & \\ & \mathbf{x} \in X \end{aligned}$$

- Now the original problem is transformed into the following relaxed problem:

$$\begin{aligned} \text{Minimize} \quad & \mathbf{c}^T\mathbf{x} + \mu^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \\ \text{subject to} \quad & \\ & \mathbf{x} \in X \end{aligned}$$



- Now the task is to find the greatest lower bound for the above relaxed problem.
- Other well known methods include LP relaxation, Dantzig-Wolfe decomposition, column generations, dynamic programming, Benders' decomposition, etc.

Hybrid methods – new potential?

- **Synergy** - bring together the bests from both paradigms - EC and math programming methods;
- Focusing more on solving problems that are practically relevant, rather than purely for new algorithms;
- Fertile grounds for **new research ideas!!**
- EC methods can leverage on the strong theoretical foundation of the exact methods, from the field of operations research;
- Exact methods can be enhanced to solve non-convex and nonlinear problems for large problem instance sizes, and can be made more robust, with fewer assumptions.

Blum, C. and Raidl, G.R. *Hybrid Metaheuristics - Powerful Tools for Optimization*. Artificial Intelligence: Foundations, Theory, and Algorithms, Springer 2016.

Blum, C., Puchinger, J. and Raidl, G.R. and Roli, A.”Hybrid metaheuristics in combinatorial optimization: A survey”. Applied Soft Computing 11(6): 4135-4151 (2011).

Constrained Pit (CPIT) problems in mining

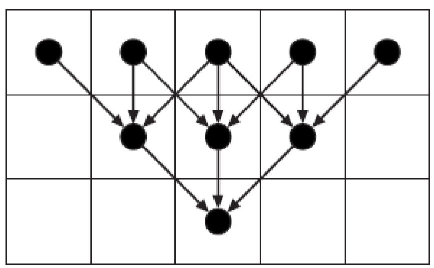
- Open-pit mining is an important industry in Australia
- Small increases, or decreases, in efficiency can have a large effect on profit.
- The two most critical tasks in planning an open pit mine is deciding what to mine, and also the order in which to mine it.
- The CPIT problem combines these two tasks, allowing the mine operator to estimate the total value of the mine over its life and also to identify the most valuable areas for excavation.
- Properties:
  - very large-scale
  - Few side constraints, but many variables and precedence constraints;
  - Current MIP solvers cannot handle without first using decomposition.

CPIT problem modelling

- The mine is divided up into discrete *blocks*, each given a value and a resource cost to extract
- The objective of the CPIT problem is to maximise the net present value (NPV) over the life of the mine, taking into account the following restrictions:
  1. Each block is mined at most once
  2. A block cannot be mined before all of its predecessors
  3. The resource limits consumed by mining blocks must not be violated

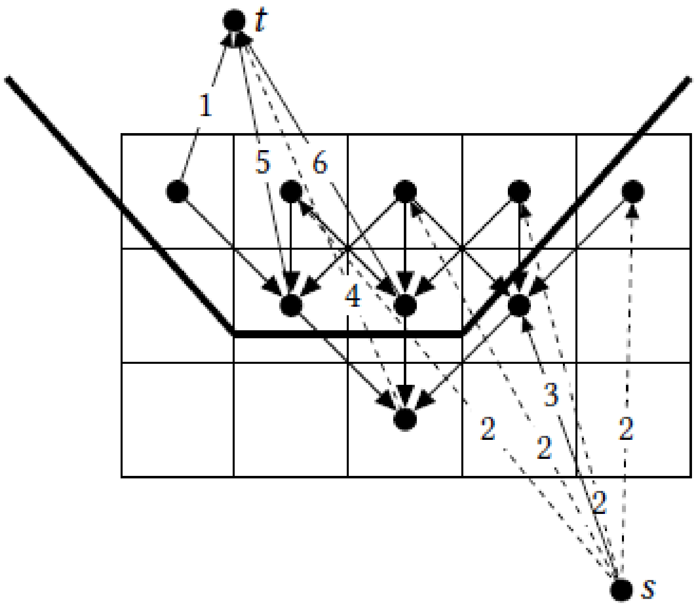
1	-2	-2	-2	-2
	5	6	-3	
		4		

(a)



(b)

Figure: A 2D cross-section of the CPIT problem is represented as a graph with precedence constraints as arcs between blocks.



The problem can be modelled as a network flow problem, and the goal is to find the maximum closure, which gives the maximum profit.

Kenny, A., Li, X. and Ernst, A.T. (2018), "A Merge Search Algorithm and its Application to the Constrained Pit Problem in Mining", in Proceedings of the 2018 Conference on Genetic and Evolutionary Computation Conference (GECCO), Kyoto, Japan, ACM, pp.316 - 323, 2018.
Kenny, A., Li, X., Ernst, A.T. and Thiruvady, D., (2017), "Towards Solving Large-Scale Precedence Constrained Production Scheduling Problems in Mining", in Proceedings of the 2017 Conference on Genetic and Evolutionary Computation Conference (GECCO), Berlin, Germany, ACM, pp.1137-1144, 2017.

CPIT problem modelling

- The MIP formulation for the CPIT problem is given below:

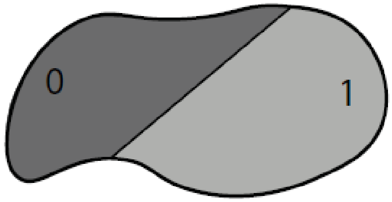
$$
\begin{aligned}
\max \quad & \sum_{b \in B} \sum_{t \in T} p_{bt}(x_{bt} - x_{bt-1}), & \\
\text{s.t.,} \quad & x_{bt} \leq x_{at} & \forall\, (a,b) \in \mathcal{P}, t \in T, \\
& x_{bt} \leq x_{b,t+1} & \forall\, b \in B, t \in T, \\
& \sum_{b \in B} q_{br}(x_{bt} - x_{bt-1}) \leq \overline{R}_{rt} & \forall\, r \in R, t \in T, \\
& x_{bt} \in \{0,1\} & \forall\, b \in B, t \in T.
\end{aligned}
$$

Here, B is the set of blocks; T the set of periods; p_{bt} the profit made from mining block b at time t; $\mathcal{P}$ is the set of precedences, where $(a,b) \in \mathcal{P}$ if block a must be mined immediately before block b; R is the set of resources; q_{br} the amount of resource r consumed by mining b; $\overline{R}_{rt}$ the total amount of r available at t; and, x_{bt} is a binary decision variable that is 1 if b is mined at t or earlier and 0 otherwise.

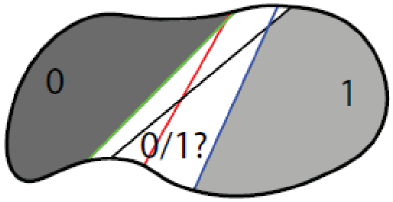
Kenny, A., Li, X. and Ernst, A.T. (2018), "A Merge Search Algorithm and its Application to the Constrained Pit Problem in Mining", in Proceedings of the 2018 Conference on Genetic and Evolutionary Computation Conference (GECCO), Kyoto, Japan, ACM, pp.316 - 323, 2018.

Merge search – from a population perspective

Principle: if a variable takes the same value across many solutions to the same problem, then it is likely to take the same value again (if another solution is generated). As the population size increases, the probability of this being true also increases. The nature of CPIT problem makes it a perfect fit to test this idea.



(a) Single solution



(b) Multiple solutions overlaid

Figure 1: By overlaying multiple solutions, regions of fixed and free variables can be identified to produce a reduced version of the problem.

- The merge operation is **a problem reduction technique** that identifies groups of variables that can be removed from main problem.
- Once the merge has occurred, a MIP solver can be used to find a solution to this restricted problem.
- This solution can then be used to generate another population of neighbouring solutions, and the cycle continues.

Kenny, A., Li, X. and Ernst, A.T. (2018), "A Merge Search Algorithm and its Application to the Constrained Pit Problem in Mining", in Proceedings of the 2018 Conference on Genetic and Evolutionary Computation Conference (GECCO), Kyoto, Japan, ACM, pp.316 - 323, 2018.

Merge search – from a population perspective

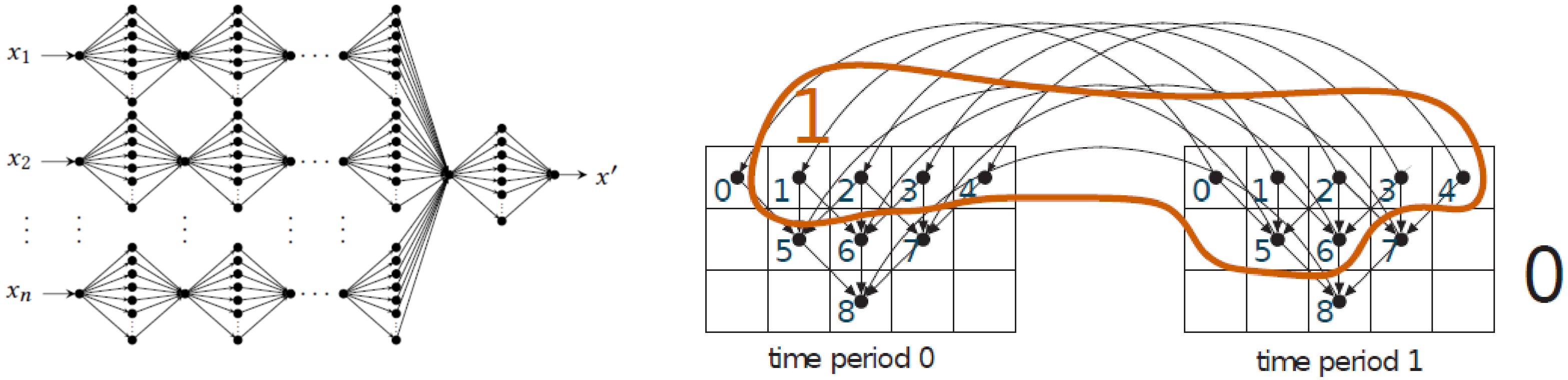
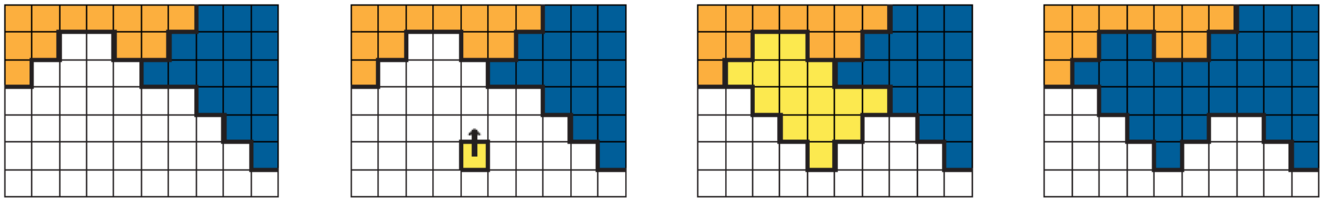


Figure: a) Parallel merge search; b) Time expanded problem graph for two time periods. The cumulative variables ensure that once a block is mined, it stays mined in subsequent periods.

- Each block has a *cone* of precedence blocks that must be mined before it
- A population of solutions is produced by swapping these cones of blocks between periods



(a) Initial solution (b) Block and direction selected (c) Block cone computed (d) Swap period of blocks in cone

Figure: A block and its predecessor cone is swapped to create a neighbouring solution.

Kenny, A., Li, X. and Ernst, A.T. (2018), "A Merge Search Algorithm and its Application to the Constrained Pit Problem in Mining", in Proceedings of the 2018 Conference on Genetic and Evolutionary Computation Conference (GECCO), Kyoto, Japan, ACM, pp.316 - 323, 2018.

Merge search – from a population perspective

Table 1: Characteristics of *minelib* [7] datasets.

Instance	Blocks	Precedences	Periods	Variables	Constraints
newman1	1,060	3,922	6	6,360	29,904
zuck_small	9,400	145,640	20	188,000	3,100,840
kd	14,153	219,778	12	169,836	2,807,196
zuck_medium	29,277	1,271,207	15	439,155	19,507,290
marvin	53,271	650,631	20	1,065,420	14,078,080
zuck_large	96,821	1,053,105	30	2,904,630	34,497,840

Initial solution construction: cones of blocks are computed and ranked according to their value and resource usage and then mined heuristically until the resource limits were reached for each period.

Table 2: Results on *minelib* dataset instances. Mean and standard deviation not reported for *minelib* [7], as the only information given was a single objective value. The subscripts $_{RS}$ and $_{CS}$ stand for random search and cone search respectively, and denote the algorithm used to construct the initial solutions.

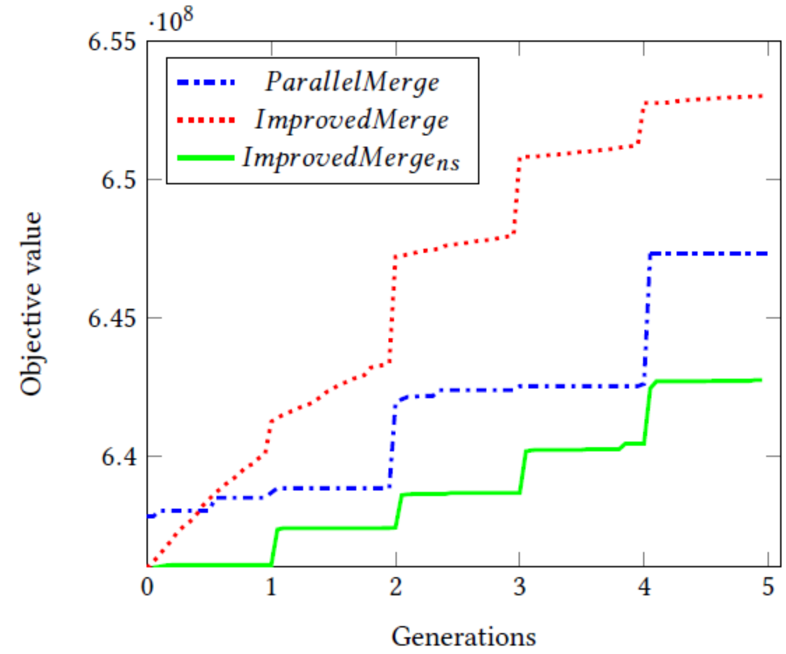
	LP UB	*minelib*	*RandomSearch*		*ConeSearch*		*SerialMerge$_{RS}$*		*SerialMerge$_{CS}$*		*ParallelMerge$_{CS}$*	
Instance			mean	std. dev	mean	std. dev	mean	std. dev.	mean	std. dev.	mean	std. dev.
newman1	2.449E+07	2.348E+07	2.322E+07	9.356E+04	2.355E+07	1.021E+05	2.411E+07	1.768E+04	2.407E+07	4.889E+04	**2.413E+07**	1.506E+04
zuck_small	8.542E+08	7.887E+08	5.676E+08	9.165E+06	7.868E+08	1.246E+07	7.832E+08	7.590E+06	8.373E+08	1.249E+06	**8.390E+08**	9.056E+05
kd	4.095E+08	3.969E+08	3.508E+08	9.307E+05	3.915E+08	1.273E+06	3.845E+08	7.837E+05	3.987E+08	9.442E+05	**4.007E+08**	3.339E+06
zuck_medium	7.106E+08	6.154E+08	4.587E+08	4.405E+06	6.083E+08	1.494E+07	6.390E+08	6.524E+06	6.444E+08	2.444E+06	**6.473E+08**	1.828E+06
marvin	8.639E+08	8.207E+08	5.925E+08	9.839E+06	7.985E+08	6.507E+06	7.955E+08	4.420E+06	8.470E+08	1.422E+06	**8.500E+08**	9.534E+05
zuck_large	5.739E+07	**5.678E+07**	4.136E+07	7.470E+04	5.042E+07	2.073E+05	4.616E+07	1.224E+05	5.102E+07	2.493E+05	5.182E+07	3.196E+05

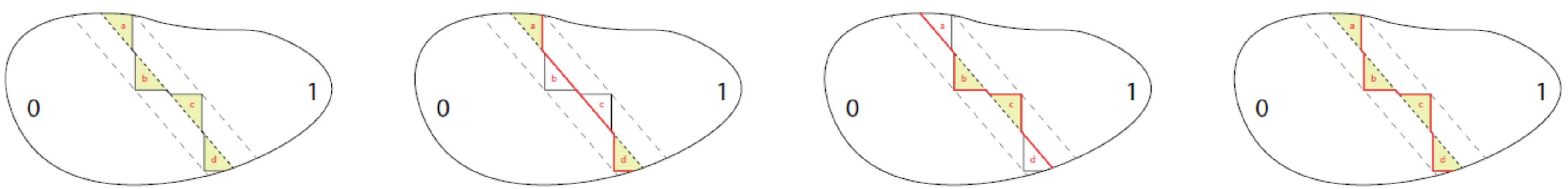
Kenny, A., Li, X. and Ernst, A.T. (2018), "A Merge Search Algorithm and its Application to the Constrained Pit Problem in Mining", in Proceedings of the 2018 Conference on Genetic and Evolutionary Computation Conference (GECCO), Kyoto, Japan, ACM, pp.316 - 323, 2018.

Further improvement on merge search

Table 3: Difference in problem size between the original and the reduced sub-problem produced by the merge operation.

Instance	Original	*ParallelMerge*		*ImprovedMerge*		
		Reduced	%	Reduced	Covered	%
newman1	6.36E+03	1.16E+03	18.2	5.98E+02	1.27E+03	19.9
zuck_small	1.88E+05	1.50E+04	8.0	7.55E+02	1.98E+03	1.1
kd	1.70E+05	1.42E+04	8.4	7.07E+02	3.37E+03	2.0
zuck_medium	4.49E+05	3.99E+04	9.1	1.74E+03	9.41E+03	2.1
marvin	1.07E+06	1.38E+04	1.3	6.52E+02	1.90E+03	0.2
zuck_large	2.90E+06	1.88E+05	6.5	5.31E+02	1.65E+03	0.06



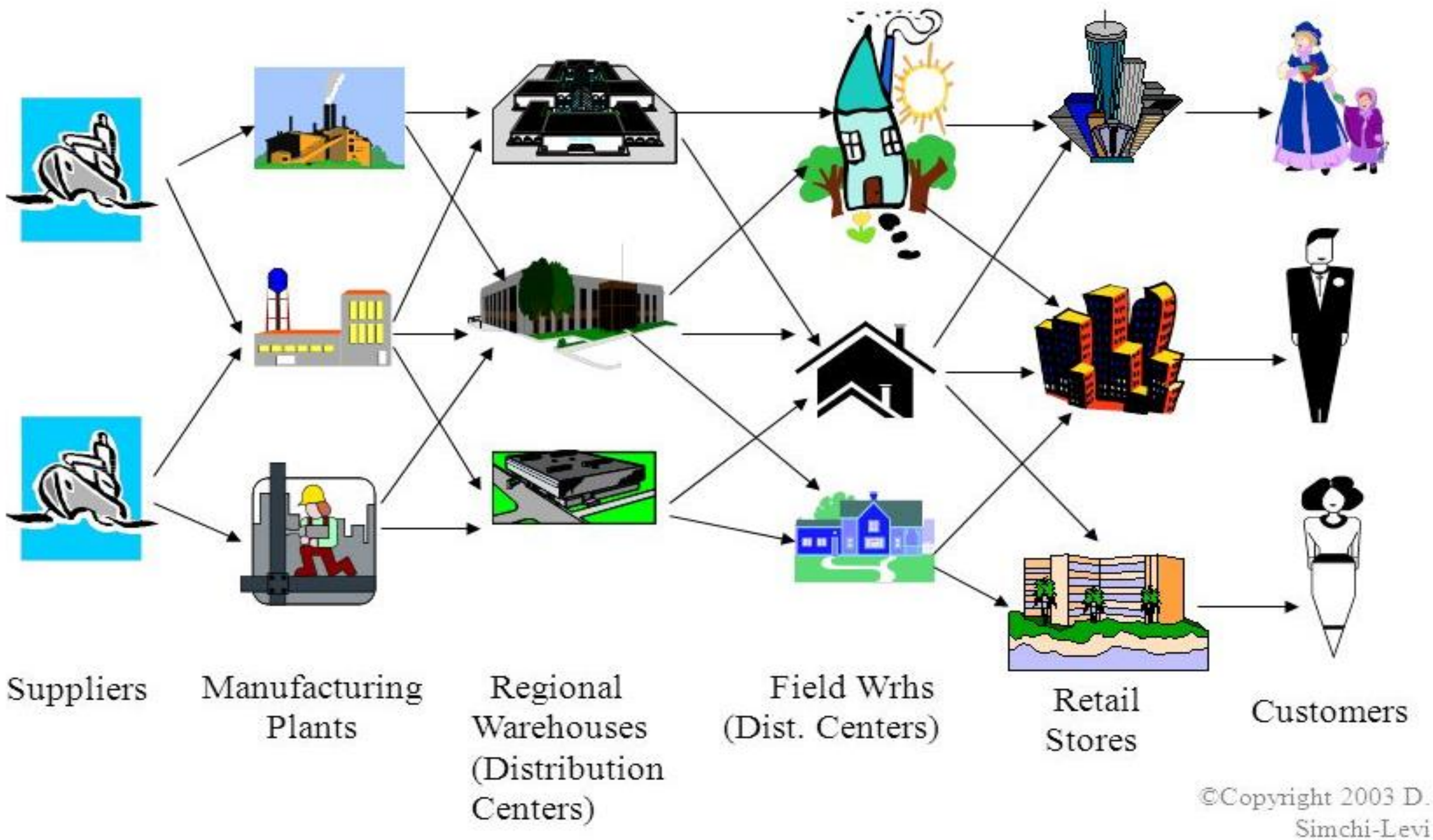


(a) Solutions $\{a, b\}$ and $\{c, d\}$ (b) Solution with groups $\{a, d\}$ (c) Solution with groups $\{b, c\}$ (d) Solution with groups $\{a, b, c, d\}$

Figure 4: By selecting different combinations of variable groups in the reduced sub-problem, new solutions can be generated.

Kenny, A., Li, X., Ernst, A.T. and Sun, Y., (2019), "An Improved Merge Search Algorithm For the Constrained Pit Problem in Open-pit Mining", in Proceedings of the 2019 Conference on Genetic and Evolutionary Computation Conference (GECCO), Prague, 2019 (accepted on 21/03/2019).

Minimum cost network flow problems



Minimum cost integer flow problem (MCFP)

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} x_{ij} \quad (1)$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = \begin{cases} q, & \text{if } i = 1 \\ 0, & \text{if } i = (2, 3, \ldots, n-1) \\ -q, & \text{if } i = n \end{cases} \quad (2)$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad (i, j = 1, \ldots, n) \quad (3)$$

$$x_{ij} \in \mathbb{Z} \quad (i, j = 1, \ldots, n) \quad (4)$$

Nodes:	1	2	3	4	5
Priorities	5	4	1	3	2

Figure 2: A chromosome for priority based encoding method.

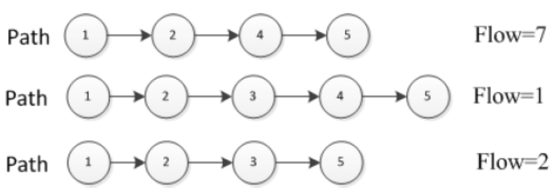
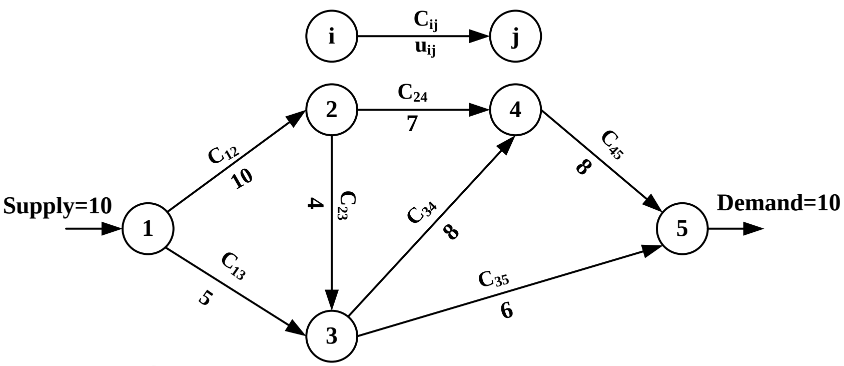
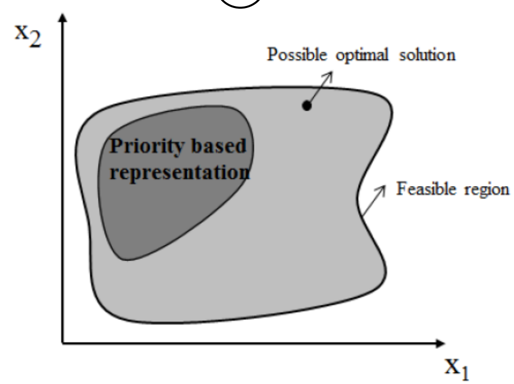
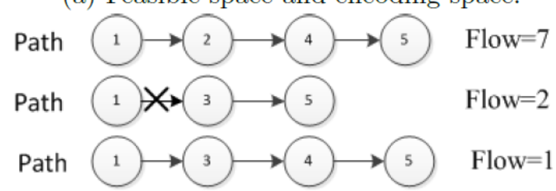


Figure 3: An example of priority based decoding paths





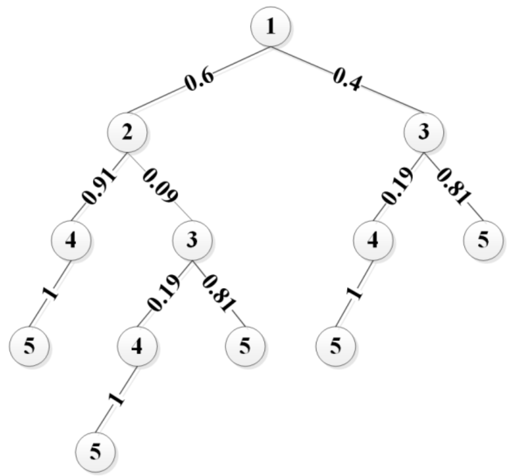
(a) Feasible space and encoding space.

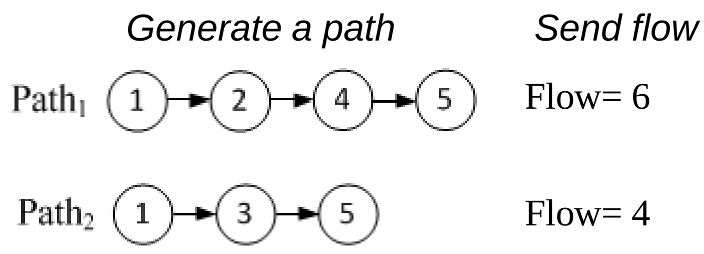


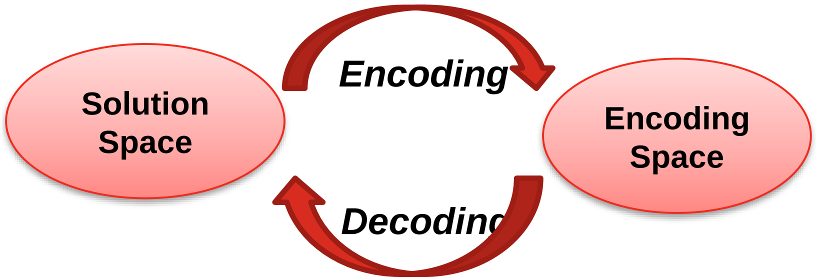
(b) A feasible solution that the priority based encoding cannot represent

Ghasemishabankareh, B., Ozlen, M., Neumann, F. and Li, X. (2018), "A Probabilistic Tree-Based Representation for Non-convex Minimum Cost Flow Problems", in Proceedings of the 15th International Conference on Parallel Problem Solving from Nature (PPSN'2018), LNCS, Springer, Coimbra, Portugal, pp.69 - 81, 2018.

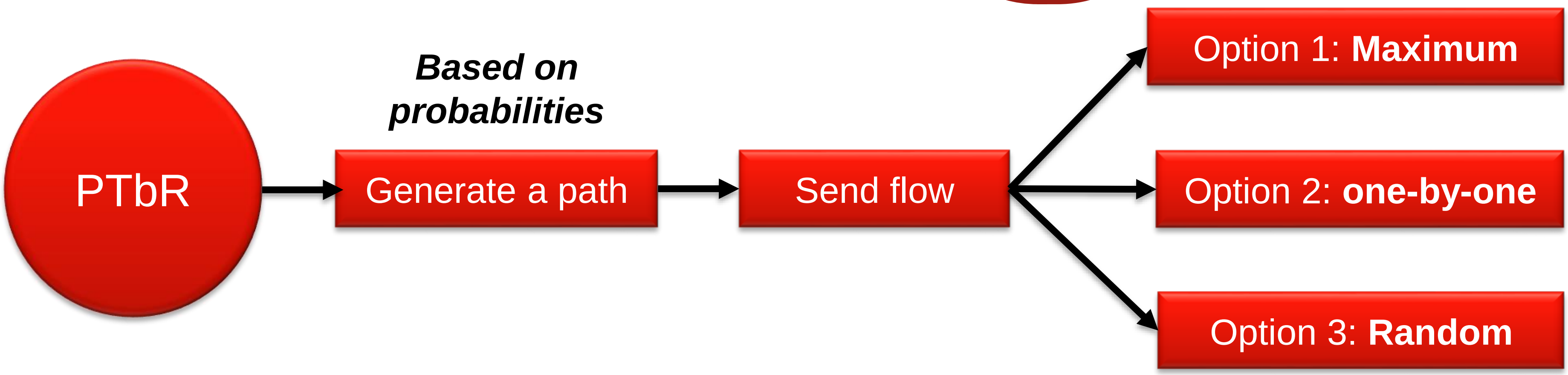
Probability tree-based representation (PTbR)





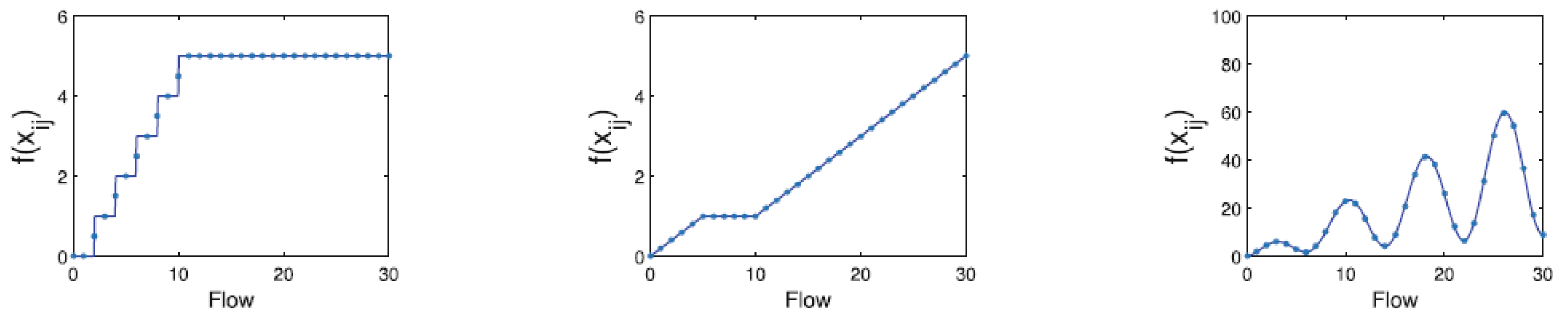


- Using GA to **evolve** representation schemes for solving MCFP;
- Using non-convex cost functions.



Ghasemishabankareh, B., Ozlen, M., Neumann, F. and Li, X. (2018), "A Probabilistic Tree-Based Representation for Non-convex Minimum Cost Flow Problems", in Proceedings of the 15th International Conference on Parallel Problem Solving from Nature (PPSN'2018), LNCS, Springer, Coimbra, Portugal, pp.69 - 81, 2018.

Minimum cost integer flow problem (MCFP)



(a) Cost function F_1 (b) Cost function F_2 (c) Cost function F_3

Table 1. A set of 35 randomly generated single-source single-sink MCFP instances.

No.	*n*	*m*	*No.*	*n*	*m*	*No.*	*n*	*m*	*No.*	*n*	*m*	*No.*	*n*	*m*	*No.*	*n*	*m*	*No.*	*n*	*m*
1		8	6		24	11		114	16		369	21		1484	26		3419	31		4882
2		8	7		34	12		98	17		385	22		1406	27		3166	32		4718
3	**5**	8	8	**10**	32	13	**20**	105	18	**40**	373	23	**80**	1560	28	**120**	3326	33	**160**	4986
4		9	9		27	14		99	19		406	24		1353	29		3212	34		4835
5		8	10		29	15		101	20		406	25		1526	30		2911	35		5130

A set of 35 single-source single sink MCFP instances is randomly generated with different number of nodes (n = {5, 10, 20, 40, 80, 120, 160}). Each instance has n nodes and m arcs. Five different networks are randomly generated for each node size n.

Ghasemishabankareh, B., Ozlen, M., Neumann, F. and Li, X. (2018), "A Probabilistic Tree-Based Representation for Non-convex Minimum Cost Flow Problems", in Proceedings of the 15th International Conference on Parallel Problem Solving from Nature (PPSN'2018), LNCS, Springer, Coimbra, Portugal, pp.69 - 81, 2018.

Minimum cost integer flow problem (MCFP)

No.	n	m	PtGA-R			PtGA-O			PTGA-M			PrGA			LINDOGlobal		AlphaECP		h
			t	mean	std	t	mean	std	t	mean	std	t	mean	std	t'	OBJ	t'	OBJ	
1	5	8	5	**30.1752**	7.29E-15	12	**30.1752**	7.29E-15	3	**30.1752**	7.29E-15	7	**30.1752**	7.29E-15	1	**30.1752**	1	**30.1752**	0
2		8	5	**32.2126**	0.00E+00	13	**32.2126**	0.00E+00	3	**32.2126**	0.00E+00	7	**32.2126**	0.00E+00	2	**32.2126**	1	**32.2126**	0
3		8	5	**33.0507**	7.29E-15	15	**33.0507**	7.29E-15	4	**33.0507**	7.29E-15	6	**33.0507**	7.29E-15	1	**33.0507**	1	**33.0507**	0
4		9	5	**33.1016**	7.29E-15	13	**33.1016**	7.29E-15	4	**33.1016**	7.29E-15	7	**33.1016**	7.29E-15	1	**33.1016**	1	**33.1016**	0
5		8	6	**40.3756**	2.19E-14	15	**40.3756**	2.19E-14	3	**40.3756**	2.19E-14	6	**40.3756**	2.19E-14	1	**40.3756**	1	**40.3756**	0
6	10	24	45	29.2974	2.08E-01	76	29.1682	8.03E-02	42	29.2961	9.99E-02	32	30.0343	3.16E-05	3600	**29.135**	268	30.021	-1
7		34	57	19.8109	2.03E-01	88	20.1956	3.09E-01	45	20.7775	2.94E-01	42	24.1056	6.01E-01	3600	**19.5957**	3600	20.184	-1
8		32	48	23.4834	1.16E-01	85	23.6681	1.46E-01	37	24.0765	2.34E-01	28	25.3353	1.39E-01	3600	**23.3061**	350	24.018	-1
9		27	48	26.1797	2.31E-01	69	26.2407	1.52E-01	34	26.4196	7.96E-02	29	28.3965	2.01E-01	3600	**25.9165**	525	25.929	-1
10		29	49	19.8438	9.12E-02	68	20.2202	2.30E-01	34	21.4404	1.55E-01	32	23.3819	2.95E-01	3600	**19.6325**	740	**19.6325**	-1
11	20	114	160	8.8767	4.60E-01	172	10.9784	4.74E-01	178	13.0274	7.23E-01	153	15.8245	2.03E-01	3600	8.3929	3600	**7.0800**	-1
12		98	141	11.7898	2.18E-01	167	12.4354	3.02E-01	156	13.2006	4.27E-01	160	15.4705	2.40E-01	3600	11.5964	3600	**11.372**	-1
13		105	213	8.2120	4.40E-01	239	9.619	5.36E-01	191	10.7931	4.36E-01	133	11.7116	4.25E-01	3600	7.0803	3600	**6.4935**	-1
14		99	187	**10.1773**	3.85E-01	227	11.3854	7.18E-01	202	12.881	6.11E-01	175	15.1102	9.30E-01	3600	10.5551	3600	10.7775	1
15		101	132	14.9139	3.17E-01	191	15.5517	5.14E-01	104	16.5822	7.43E-01	154	18.4694	5.16E-01	3600	13.8717	3600	**13.6730**	-1
16	40	369	340	1.6119	3.08E-01	437	3.1972	4.35E-01	362	4.1695	5.34E-01	411	6.6595	7.00E-01	3600	**0.4433**	3600	5.366	-1
17		385	316	**3.5652**	4.57E-01	376	4.853	4.95E-01	285	6.0036	3.92E-01	407	10.1381	8.84E-01	3600	4.9674	3600	10.375	1
18		373	393	0.5091	3.26E-01	523	1.9296	4.46E-01	406	3.0951	7.99E-01	432	5.1361	1.09E+00	3600	**0.1937**	3600	4.717	-1
19		406	330	0.8437	3.93E-01	370	3.22	6.36E-01	279	6.5308	4.73E-01	342	10.684	6.18E-01	3600	**0.2118**	3600	2.298	-1
20		406	359	3.5742	4.14E-01	435	6.177	6.60E-01	320	9.1484	6.27E-01	360	12.2394	7.58E-01	3600	8.7013	3600	**2.7180**	-1
21	80	1484	336	**0.7333**	2.71E-04	401	0.8181	1.67E-01	375	2.3523	7.75E-01	459	3.2517	1.32E+00	3600	NF	3600	5.703	1
22		1406	326	**0.6737**	4.90E-04	336	0.806	1.56E-01	369	2.2178	4.64E-01	506	5.4074	1.03E+00	3600	NF	3600	4.982	1
23		1560	279	**0.8085**	3.36E-04	361	1.6541	4.84E-01	278	4.128	8.63E-01	429	5.6396	9.22E-01	3600	NF	3600	7.476	1
24		1353	342	**0.6585**	3.50E-04	464	1.793	5.88E-01	354	4.3911	7.70E-01	779	8.1006	1.32E+00	3600	NF	3600	4.18	1
25		1526	322	**0.7628**	3.39E-04	394	1.0477	3.20E-01	302	2.7758	5.56E-01	583	5.0059	1.11E+00	3600	NF	3600	8.642	1
26	120	3419	725	**1.0924**	5.37E-04	725	2.8201	3.69E-01	711	5.525	4.39E-01	877	8.5144	1.17E+00	3600	NF	3600	6.585	1
27		3166	728	**1.0818**	9.44E-04	805	2.4484	3.89E-01	624	4.9784	6.08E-01	754	7.6565	1.16E+00	3600	NF	3600	2.103	1
28		3326	892	**1.0152**	6.75E-04	893	2.3935	3.03E-01	636	5.0317	6.01E-01	906	7.5819	7.76E-01	3600	NF	3600	13.321	1
29		3212	748	**1.0532**	7.88E-04	877	2.5385	5.05E-01	673	5.2673	6.27E-01	777	10.2727	1.74E+00	3600	NF	3600	3.414	1
30		2911	774	**0.9446**	4.06E-04	828	2.0145	4.28E-01	697	4.2784	6.21E-01	817	7.5239	1.37E+00	3600	NF	3600	4.297	1
31	160	4882	837	**12.2598**	2.55E+00	914	14.5003	4.81E-01	950	15.6057	5.54E-01	922	13.8152	2.74E-01	3600	NF	3600	14.18	1
32		4718	961	**6.1413**	1.42E+00	919	15.6089	9.73E-01	952	17.0593	7.89E-01	927	14.6681	9.81E-01	3600	NF	3600	10.578	1
33		4986	902	**8.5483**	1.21E+00	912	16.2194	6.18E-01	947	17.2818	8.47E-01	1345	15.1022	7.50E-01	3600	NF	3600	14.45	1
34		4835	853	**6.3798**	1.01E+00	1064	11.199	4.81E-01	1067	12.1144	6.70E-01	942	10.3464	4.48E-01	3600	NF	3600	14.1422	1
35		5130	994	**10.6176**	1.22E+00	1068	19.7703	7.79E-01	1081	20.5212	8.59E-01	896	18.9247	5.97E-01	3600	NF	3600	15.043	1

Table: results on cost function F1. Three possible ways to send the flow over the generated path: randomly (R), one-by-one (O), or by a maximum possible amount (M).

Ghasemishabankareh, B., Ozlen, M., Neumann, F. and Li, X. (2018), "A Probabilistic Tree-Based Representation for Non-convex Minimum Cost Flow Problems", in Proceedings of the 15th International Conference on Parallel Problem Solving from Nature (PPSN'2018), LNCS, Springer, Coimbra, Portugal, pp.69 - 81, 2018.

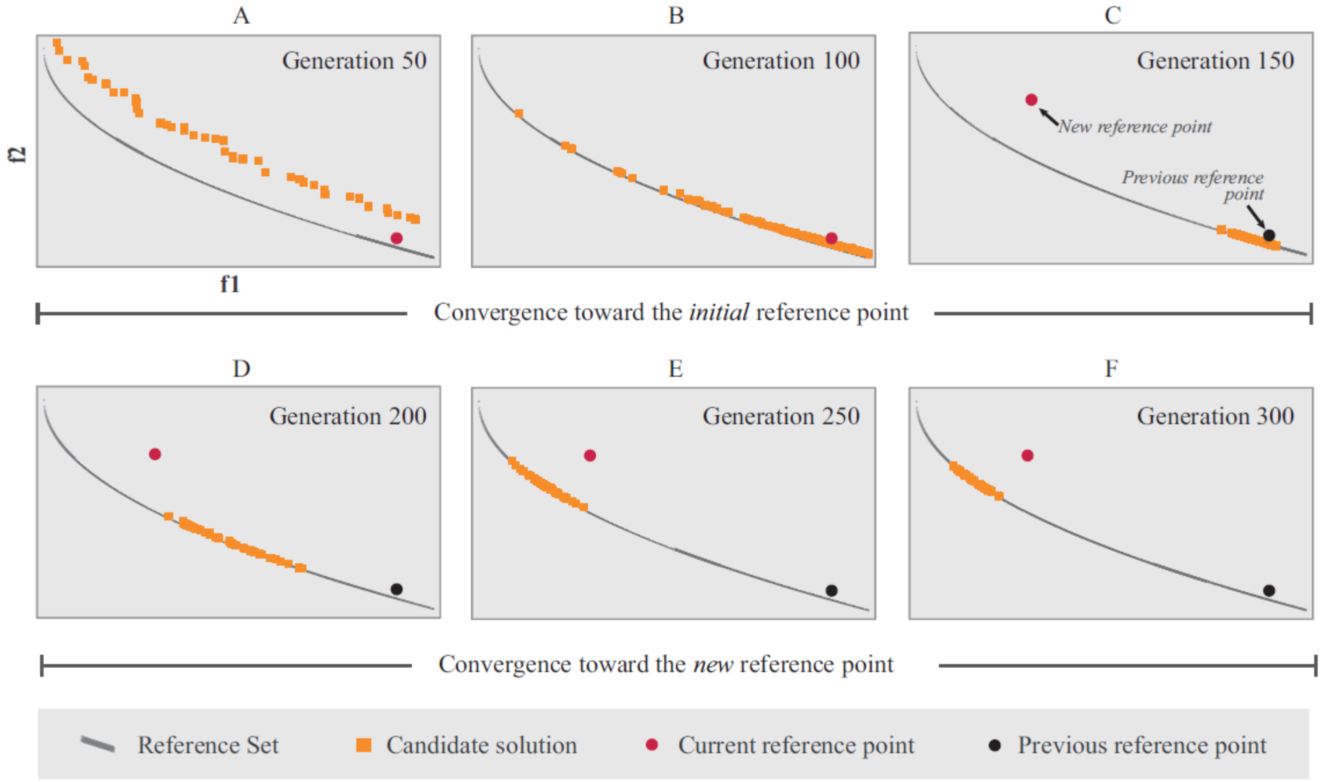
Human-in-the-loop for EMO

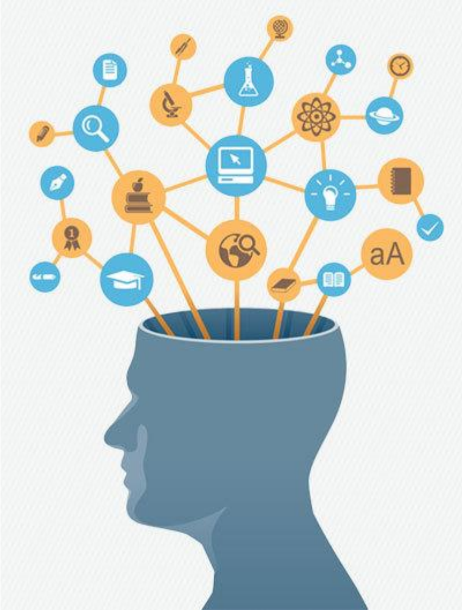
- The field of evolutionary multiobjective optimisation has traditionally involved the approximation of the entire Pareto Front of the objective space.
- The computational effort required to find these solutions is significant and the number of solutions found can be considerable.
- By incorporating user preferences the search for solutions can be directed or focussed toward a region of interest.
- It is often assumed that the user has a set of predetermined preferences and an implicit value function that can evaluate potential solutions.
- To aid preference articulation in optimisation, the technique of progressive interactivity has been incorporated from Operations Research.
- This allows the user to learn about the problem, explore options and formulate their preferences while reducing the search space and computation time

Taylor, K. and Li, X. (2018), "Interactive Multiobjective Optimisation: Preference Changes and Algorithm Responsiveness", in Proceedings of the 2018 Conference on Genetic and Evolutionary Computation Conference (GECCO), Kyoto, Japan, ACM, pp.761-768, 2018.

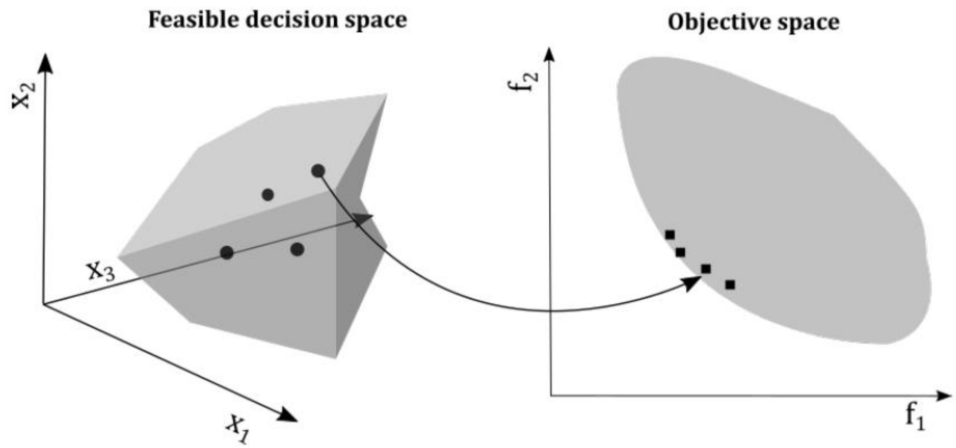
An interactive approach for EMO

- Decision maker interacts with an EMO algorithm during its optimization run
- DM can be educated and can give intermediate feedback. This means preference information can be adjusted during the run.
- The algorithm is more adaptive to changing needs.
- Search space can be significantly reduced, since effort is more targeted to regions of interests. **Machine learning** can be used to learn and model preference information.





An interactive approach for EMO



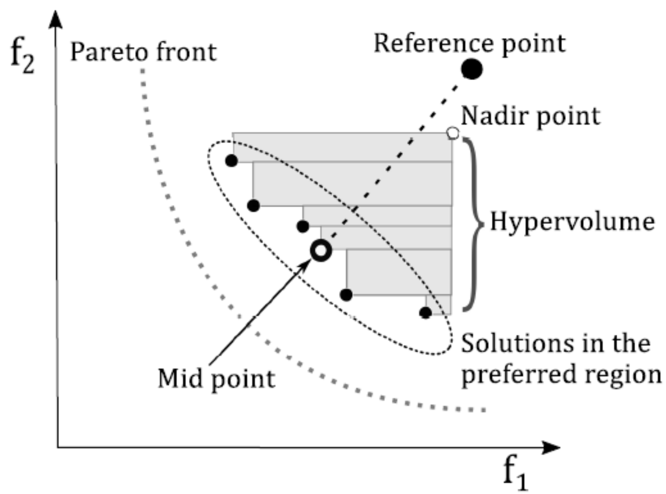
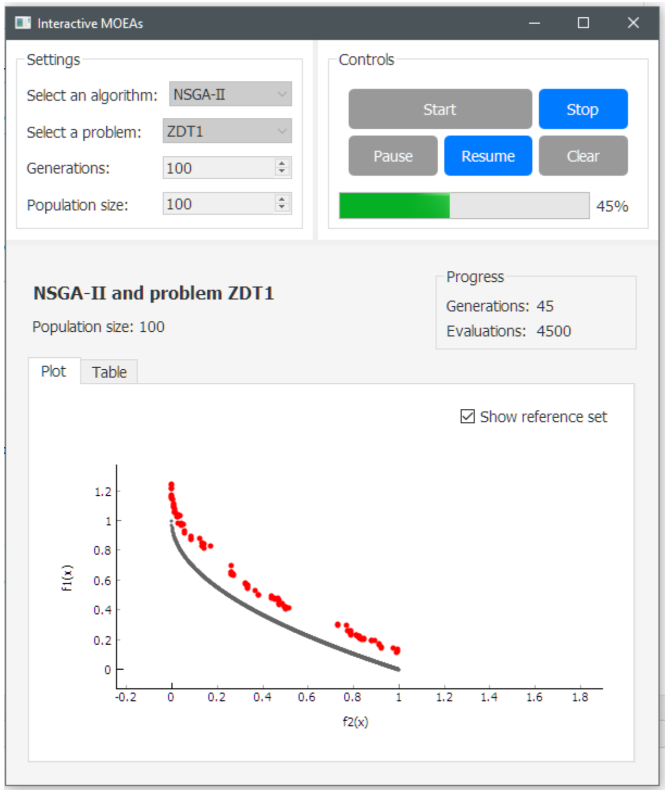


Figure 1: Preference Region hypervolume for a single generation



An interactive approach for EMO

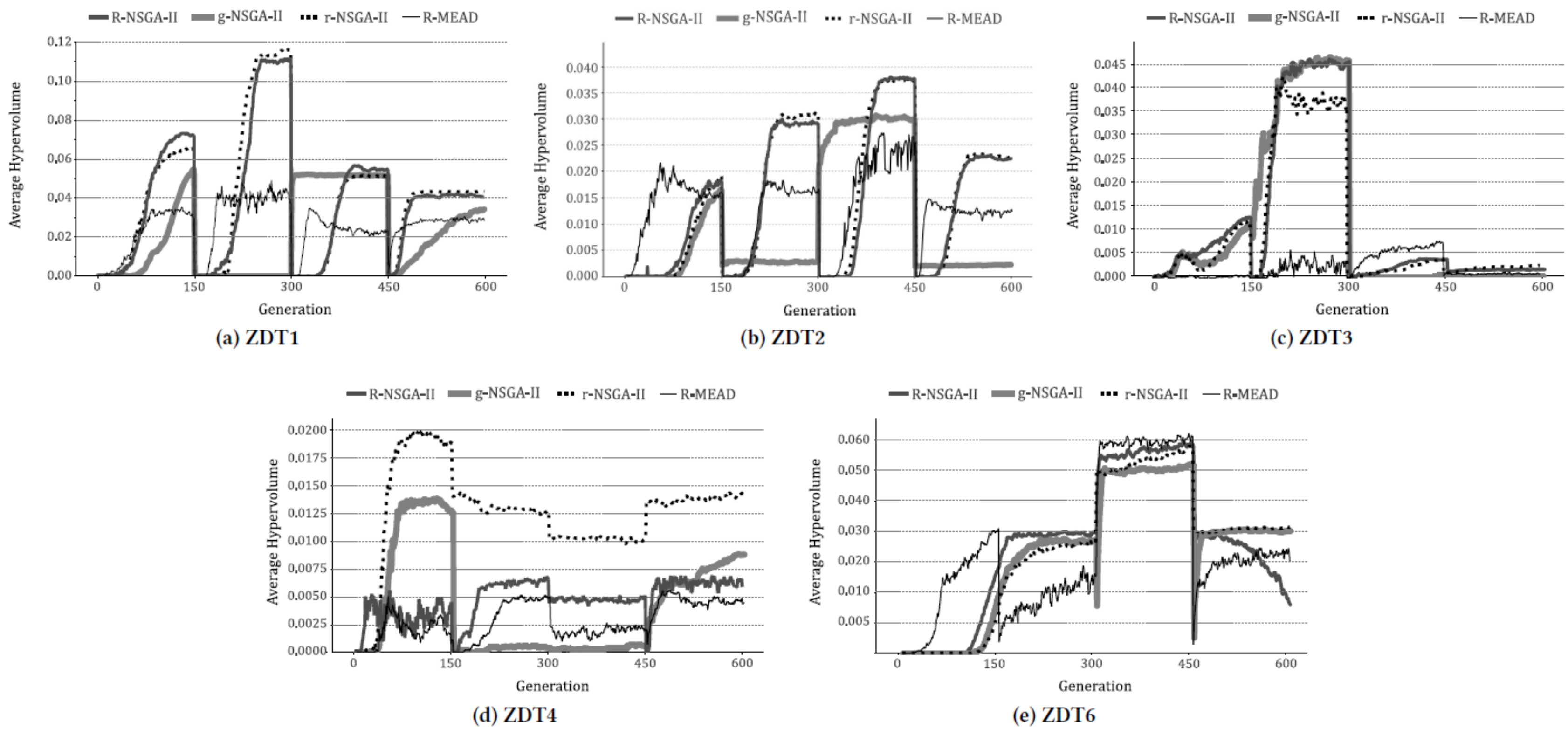


Figure 2: Average hypervolume by generation - 30 runs with problems ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6.

Taylor, K., Li, X. and Chan, J. (2019), "Improving Algorithm Response to Preference Changes in Multiobjective Optimisation Using Archives", in Proceedings of Congress of Evolutionary Computation (CEC 2019), IEEE, 2019 (accepted on 08/03/2019).
Taylor, K. and Li, X. (2018), "Interactive Multiobjective Optimisation: Preference Changes and Algorithm Responsiveness", in Proceedings of the 2018 Conference on Genetic and Evolutionary Computation Conference (GECCO), Kyoto, Japan, ACM, pp.761-768, 2018.

Truss structural optimization

Truss structural optimization

- Finding an optimal design for a truss structure involves optimizing its **topology, size, and shape**.
- A truss design problem is usually **multimodal**, meaning that the problem offers multiple optimal designs in terms of topology and/or size of the members, but they are evaluated to have similar or equally good objective function values.

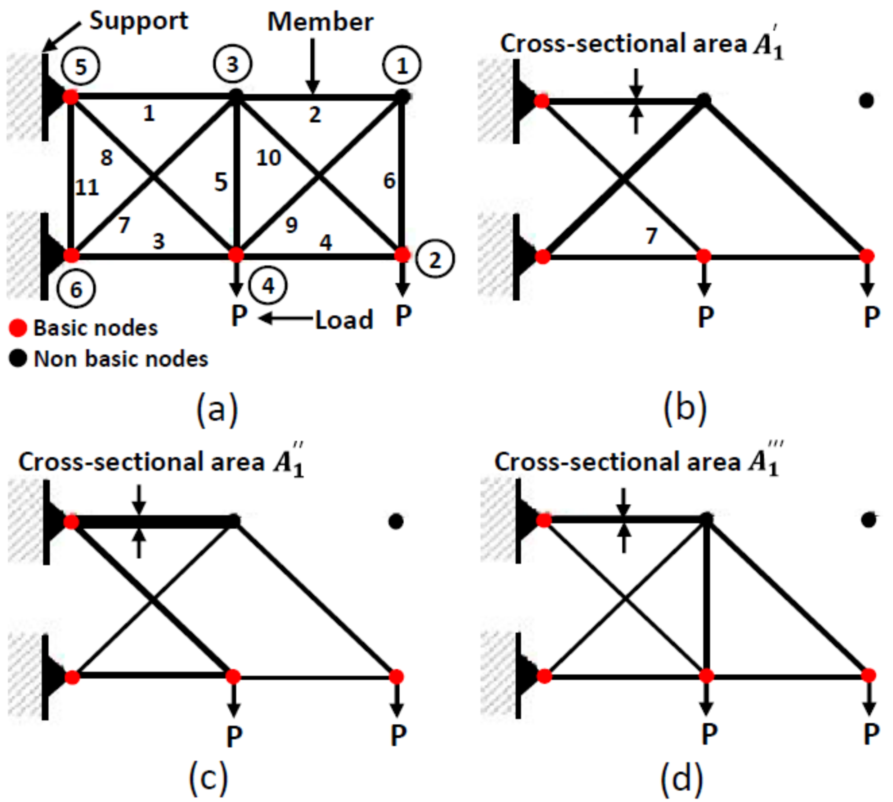


Figure: Illustration of (a) 11-member, 6-node ground structure and (b), (c), and (d) its three different design solutions.

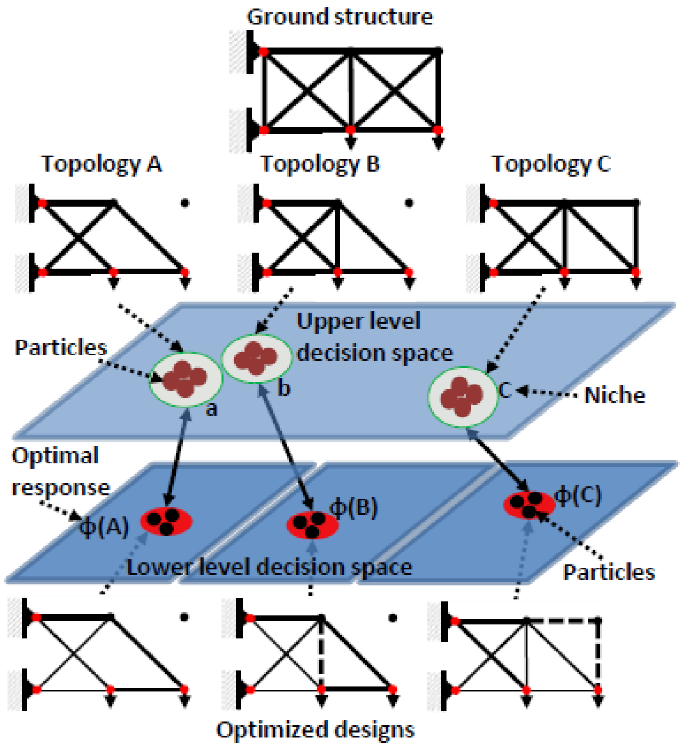
Bilevel formulation of the truss problem

A bilevel formulation for the truss problem:

- We treat the topology optimization as the upper level optimization task, and the size and shape optimization as the lower level optimization task.
- The goal is to obtain multiple truss designs by considering both its topology and size simultaneously.

$$\begin{aligned} \min_{\vec{x_u} \in X_u, \vec{x_l} \in X_l} \quad & F(\vec{x_u}, \vec{x_l}) \\ s.t. \quad & \vec{x_l} \in \underset{\vec{x_l} \in X_l}{\operatorname{argmin}} \{ f(\vec{x_u}, \vec{x_l}) : g_j(\vec{x_u}, \vec{x_l}) \leq 0, j = 1 \ldots J \} \\ & G_k(\vec{x_u}, \vec{x_l}) \leq 0, k = 1 \cdots K, \end{aligned}$$

We then can apply a **niching method** at the upper level, to obtain multiple designs in terms of topology as well as the size of the truss problem.



Islam, M.J., Li, X. and Deb, K., (2017), "Multimodal Truss Structure Design Using Bilevel and Niching Based Evolutionary Algorithms", in Proceedings of the 2017 Genetic and Evolutionary Computation Conference (GECCO), Berlin, Germany, ACM, pp.274-287, 2017.

Truss solutions found by applying niching

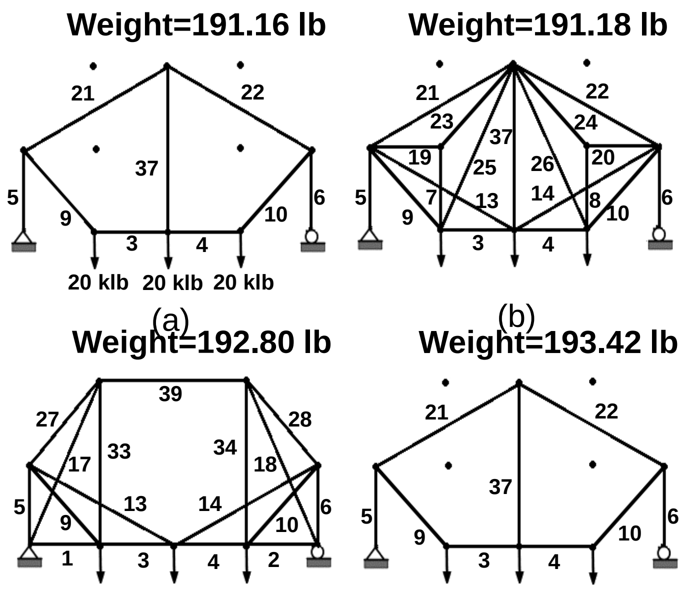


Figure: Found truss solutions for 39-member, 12-node Ground structure found by applying niching to the upper level.

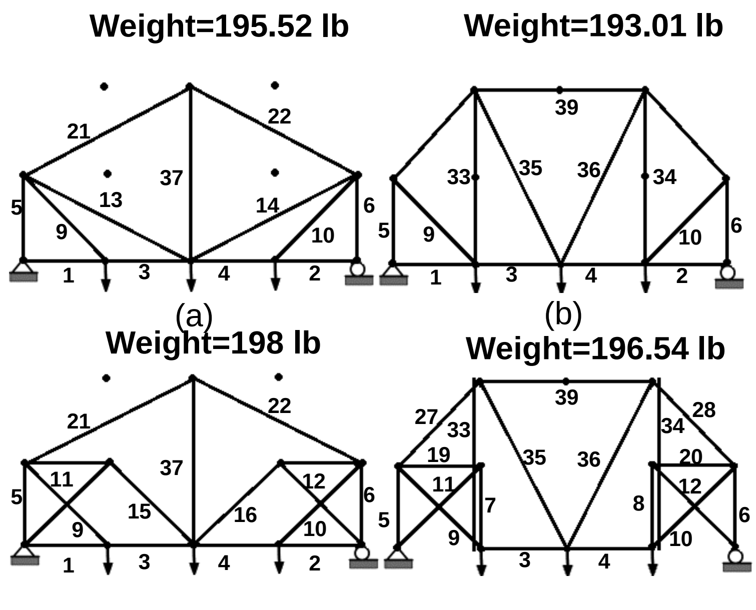


Figure: Found truss solutions using (a-b) PSO algorithms by Luh and Lin, and (c-d) GA by Deb and Gulati.

Islam, M.J., Li, X. and Deb, K., (2017), "Multimodal Truss Structure Design Using Bilevel and Niching Based Evolutionary Algorithms", in Proceedings of the 2017 Genetic and Evolutionary Computation Conference (GECCO), Berlin, Germany, ACM, pp.274-287, 2017.
Luh, G.C. and Lin, C.Y. (2011), "Optimal design of truss-structures using particle swarm optimization", Computer & Structures 89, 23–24 (2011), 2221 – 2232.
Deb, K. and Gulati, S. (2001), "Design of truss-structures for minimum weight using genetic algorithms", Finite Elements in Analysis and Design 37, 5 (2001), 447 – 465.

Take-home message …

- **Many challenges remain** when tackling real-world problems using EC methods;
- For practitioners, they are most interested in solving the problems at hand, NOT how well your methods perform on simple test functions;
- Real-world problems are far more challenging, and may require a combination of techniques in order to be effective;
- Important to study carefully the properties/characteristics of the problem under consideration, and try to incorporate the domain-specific knowledge into the design of the solution method;
- Rich ideas beyond just computer science; there are actually many others also do optimization, and we can learn a lot from them, e.g., many mature ideas in the operations research field;
- Do not be afraid of doing things differently; try NOT to follow the crowd;

Acknowledgement

- Support from our two ARC Discovery Grants (with international partner investigators Professor Kalyanmoy Deb and Professor Xin Yao).
- Professor Andreas Ernst from School of Mathematics, Monash University, for his extensive operations research expertise, and his significant role in research collaboration and PhD supervision support.
- Our PhD students, Angus Kenny, Kendall Taylor, Md Jakirul Islam, and Behrooz Ghasemishabankareh for their PhD works.
- Associate Professor Melih Ozlen (Math, RMIT), and Dr Jeffrey Chan for PhD supervision support.
- Help from Dr Yuan Sun, postdoc research fellow, supported by our 2018 ARC Discovery Grant.

Any questions?