SLS 2020 workshop
Held in conjunction with PPSN 2020

Dynastic Potential Crossover Operator

Francisco Chicano

chicano@lcc.uma.es

UNIVERSIDAD DE MÁLAGA

Joint work with

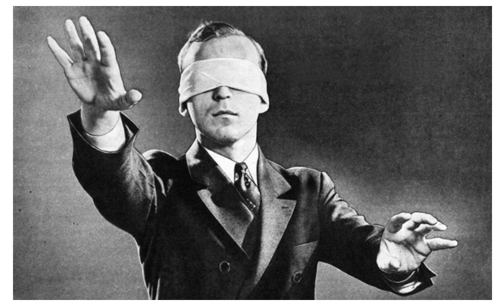
Gabriela Ochoa, Darrell Whitley and Renato Tinós

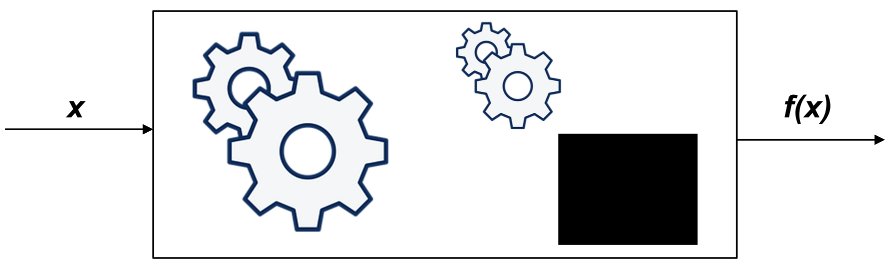
Outline

- Gray-Box Optimization
- Variable Interaction Graph (VIG)
- Recombination Operators
- Dynastic Potential Crossover
- Experimental Results
- Conclusions

Gray-Box Optimization

Gray-Box (vs. Black-Box) Optimization





For most of real problems we know (almost) all the details

Gray-Box Structure: MK Landscapes

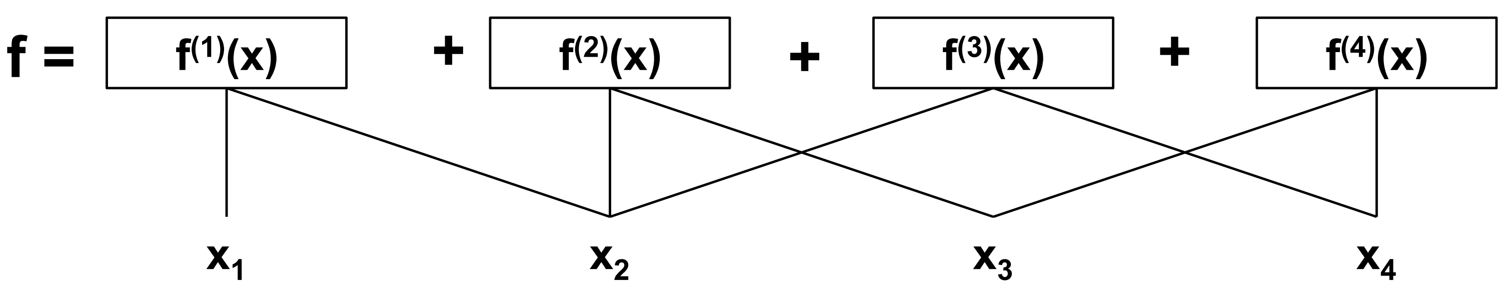
$$f(x) = \sum_{i=1}^{m} f^{(i)}(x)$$

All compresible pseudo-Boolean functions can be transformed into this in polynomial time

We focus on binary variables

Each subfunction is unknown and depends on *k* variables

Example (*k*=2):



Variable Interaction Graph

Variable Interaction

- Partial "derivative" (difference) of a pseudo-Boolean function

$$\Delta_i f(x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, 1_i, \ldots, x_n) - f(x_1, x_2, \ldots, 0_i, \ldots, x_n)$$

We say that x_i and x_j interact when $\Delta_i f$ depends on x_j

- In terms of Walsh coefficients:
  - x_i and x_j interact if there exist a nonzero Walsh coefficient with index containing both i and j

$$f = \sum_{a \in \mathbb{B}^n} w_a \psi_a.$$

Walsh expansion

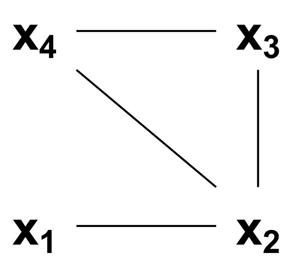
Variable Interaction Graph

- A graph where the nodes are the variables and there is an edge between two nodes if the variables interact
- Example:

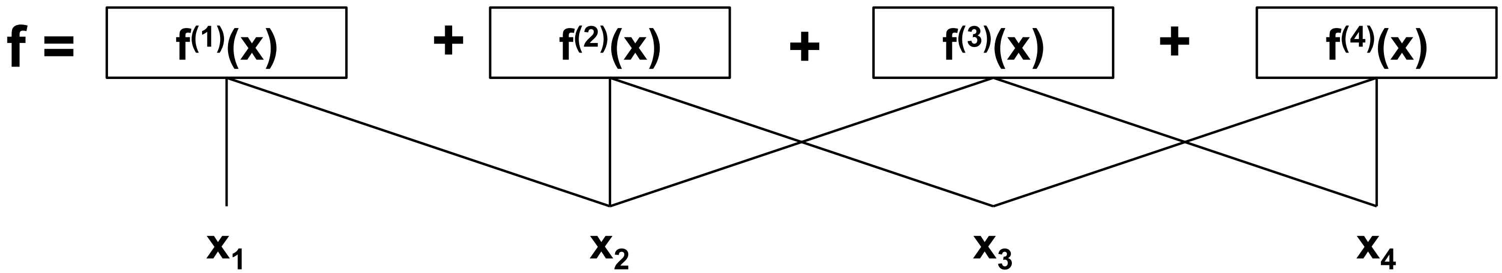
$$f(x_1, x_2, x_3, x_4) = x_1x_2 + x_2x_3 + x_2x_4 + x_3x_4$$

$$\Delta_1 f(x) = f(1,x_2,x_3,x_4) - f(0,x_2,x_3,x_4) = x_2$$

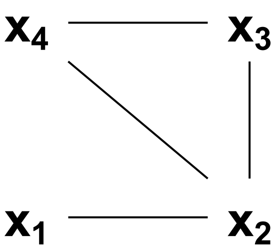
Variable Interaction Graph (VIG)



Variable Interaction Graph



We will asume that x_i and x_j interact when they appear together in the same subfunction*



Variable Interaction Graph (VIG)

If x_i and x_j don't interact: $\Delta_{ij} = \Delta_i + \Delta_j$

Recombination Operators

Recombination Operators (in EAs)

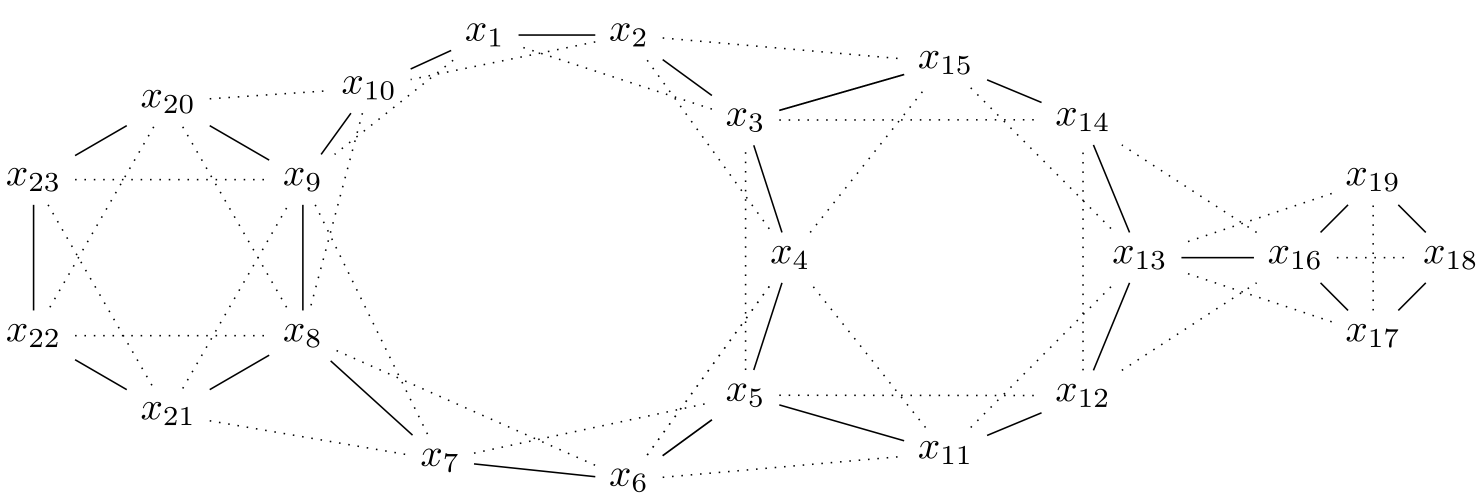
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Parent 1	a	b	b	c	d	c	c	a	a	d
Parent 2	a	a	c	c	a	b	c	b	a	d
Child	a	b	c	c	d	b	c	b	a	d

Gene Transmission → Respect Property

Radcliffe (1994)

Recombination Graph

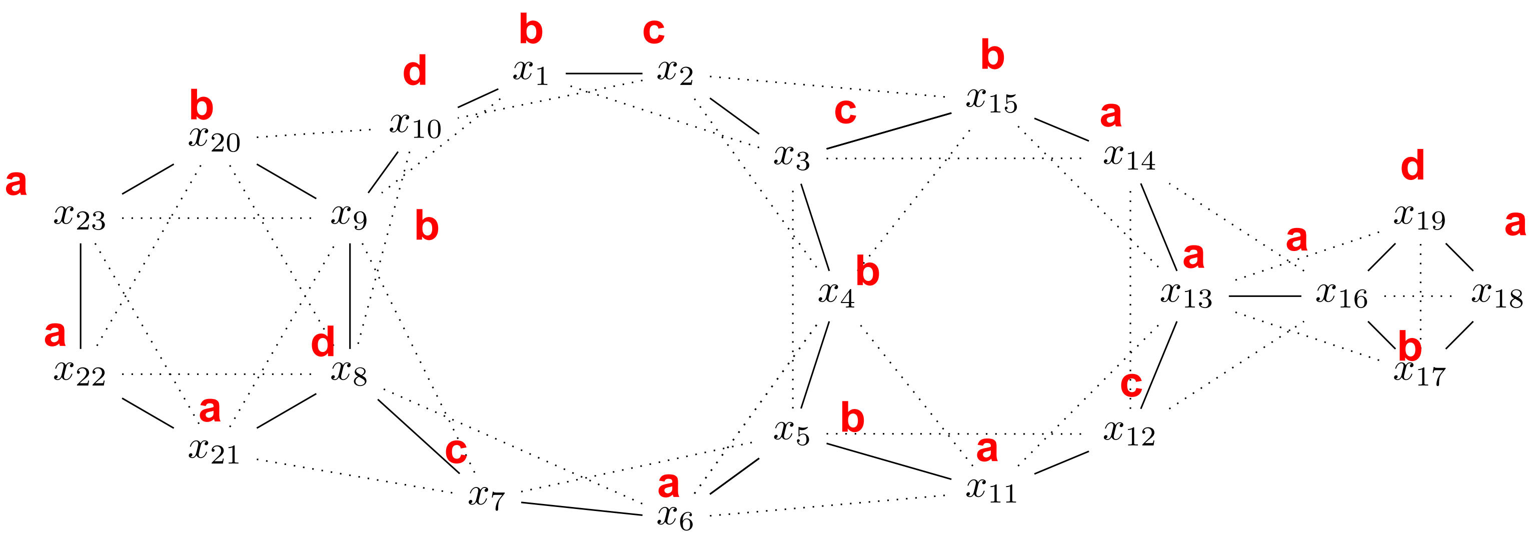
Let us suppose our function has the following VIG…



SLS 2020 workshop
Held in conjunction with PPSN 2020

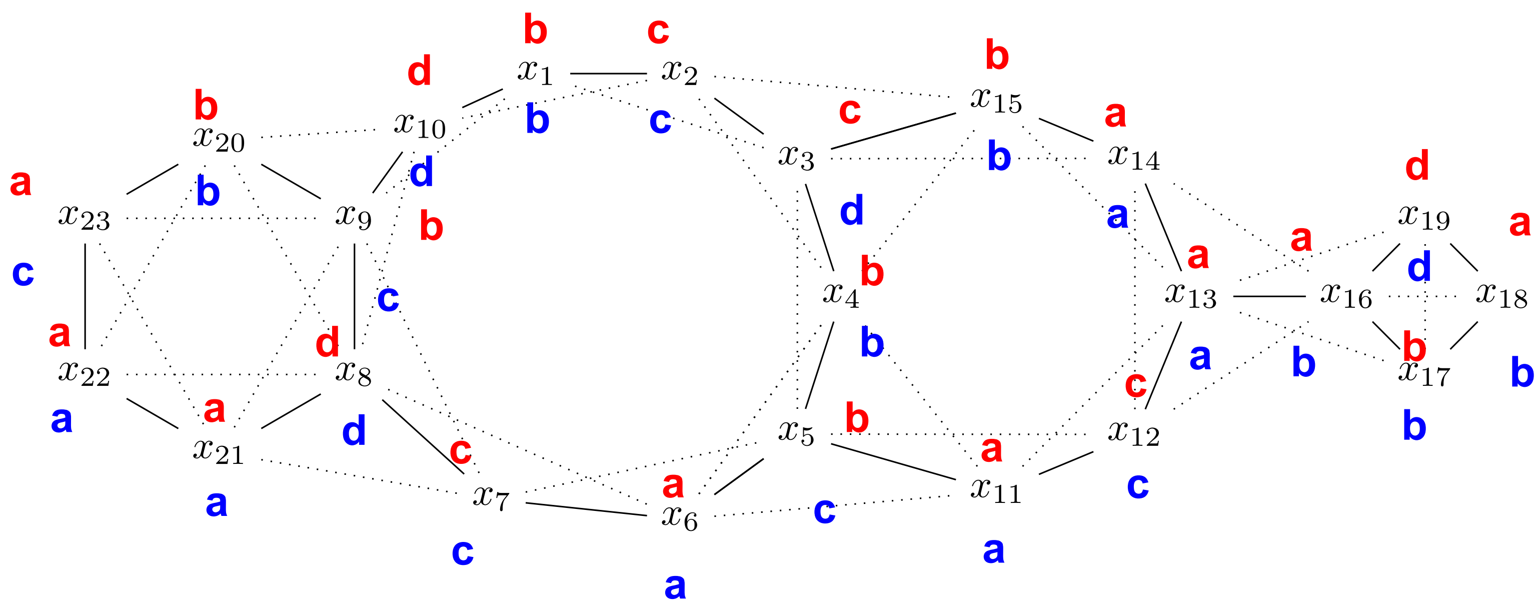
Recombination Graph

Let us suppose our function has the following VIG…



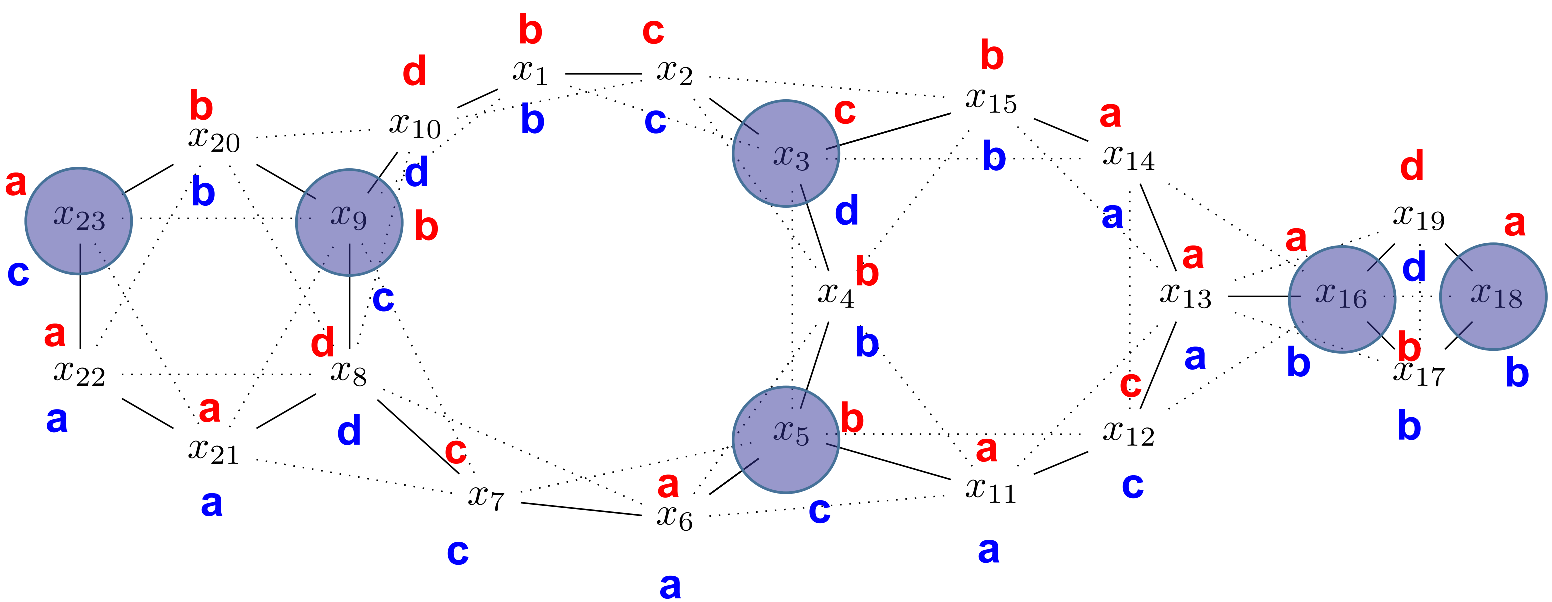
Recombination Graph

Let us suppose our function has the following VIG…



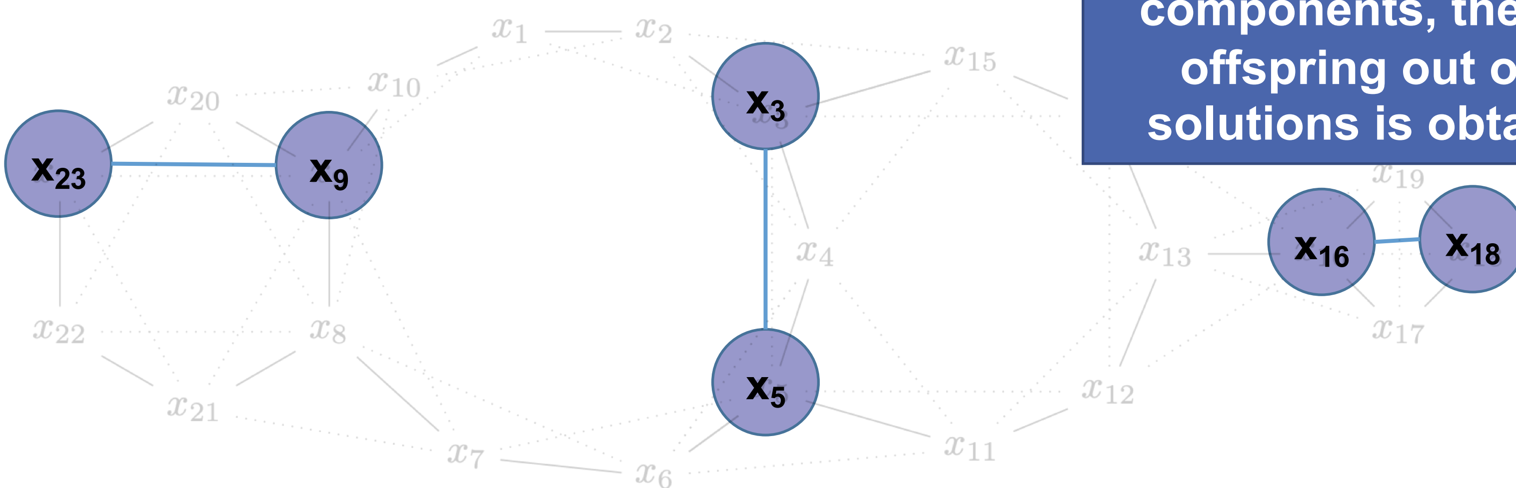
Recombination Graph

Let us suppose our function has the following VIG…



Partition Crossover

The recombination graph is a subgraph of VIG containing only the differing variables



PX: If there are *q* components, the best offspring out of 2^q solutions is obtained

Partition Crossover takes all the variables in a component from the same parent

The contribution of each component to the fitness value of the offspring is independent of each other

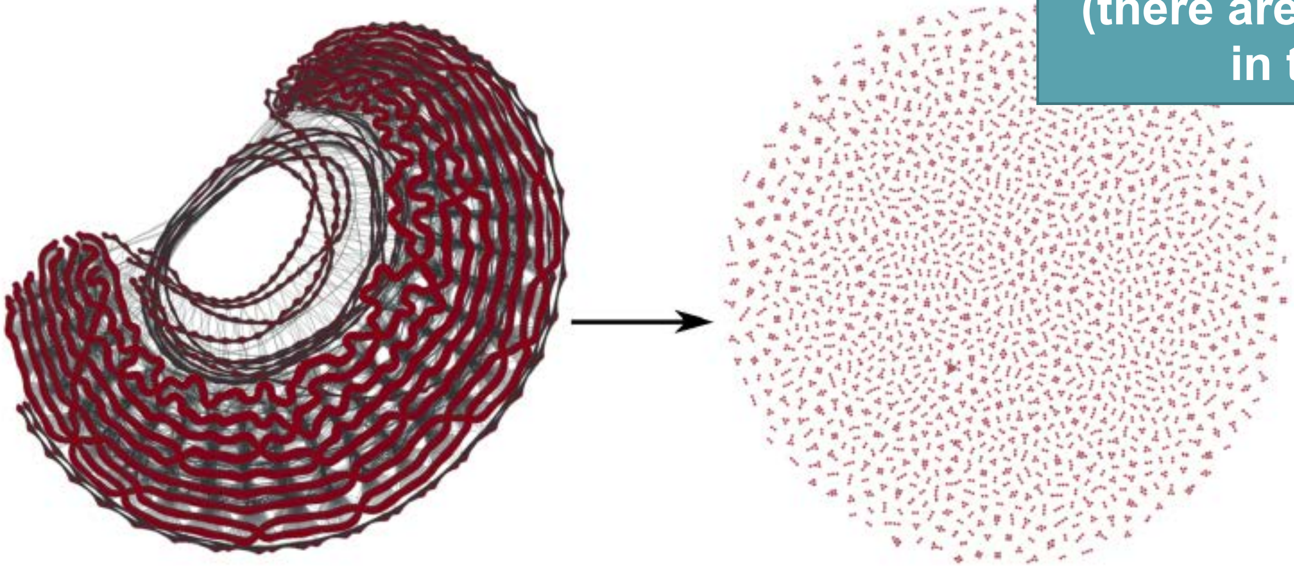
FOGA 2015: Tinós, Whitley, C.

SLS 2020 workshop
Held in conjunction with PPSN 2020

Partition Crossover

1087 components, the best out of 2^{1087} solutions obtained

(there are about 2^{366} particles in the universe)

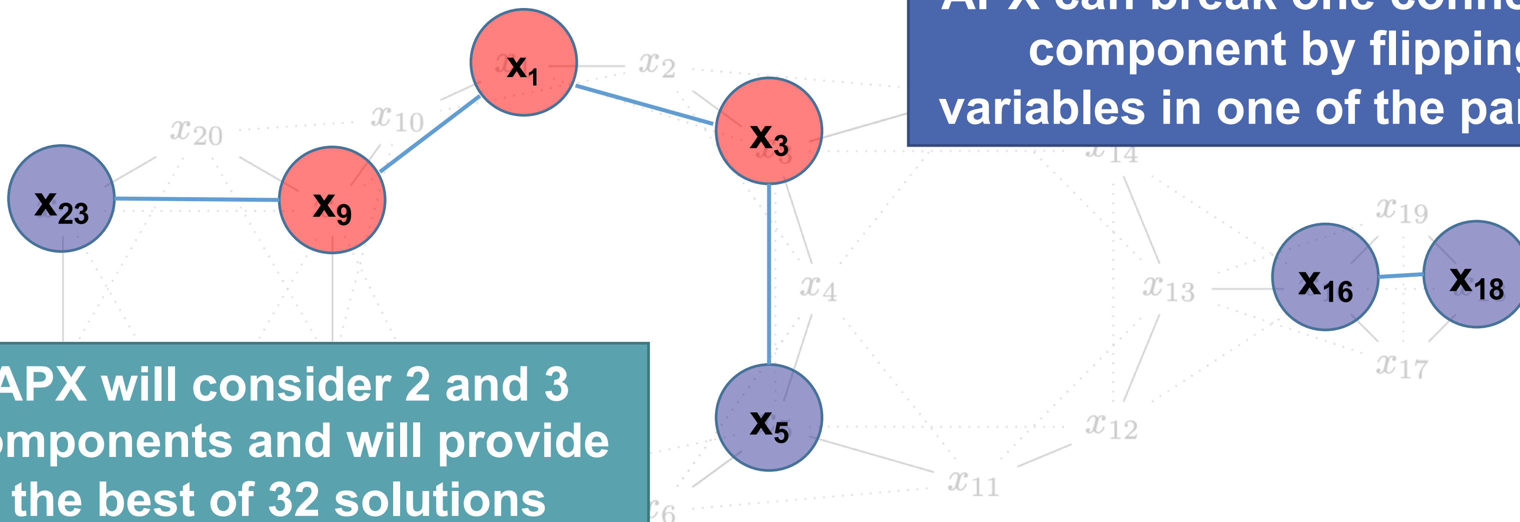


MAX-SAT instance atco_enc3_opt1_13_48 (SAT competition 2014)

EvoCOP 2017: Chen and Whitley

Articulation Points Partition Crossover

Articulation Points Partition Crossover (APX) identifies articulation points in the recombination graph



APX can break one connected component by flipping variables in one of the parents

APX will consider 2 and 3 components and will provide the best of 32 solutions

It implicitly considers all the solutions PX would consider if one or none articulation point is removed from each connected component

GECCO 2018: C., Ochoa, Whitley, Tinós

Dynastic Potential Crossover

(DPX)

Dynastic Potential

Set of solutions that can be generated by a stochastic recombination operator

If $h(x,y)$ is the Hamming distance between solutions x and y…

Single point crossover DP size: $2\,h(x,y)$

z-point crossover DP size: $O(h(x,y)^z)$ for $z << n$

Uniform crossover DP size: $2^{h(x,y)}$

Largest DP size for a recombination operator with the gene transmission property:

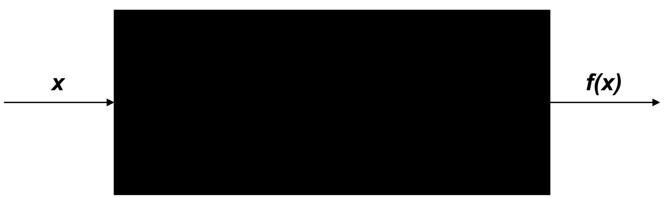
$$2^{h(x,y)}$$

Optimal Recombination Operator

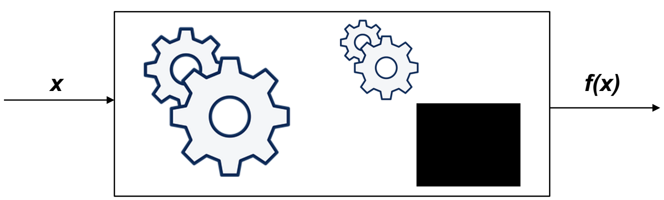
Produces the best solution in the largest dynastic potential…

Eremeev, Kovalenko (2013)

…exploring the "hyperplane" defined by the common variables ($2^{h(x,y)}$ solutions)

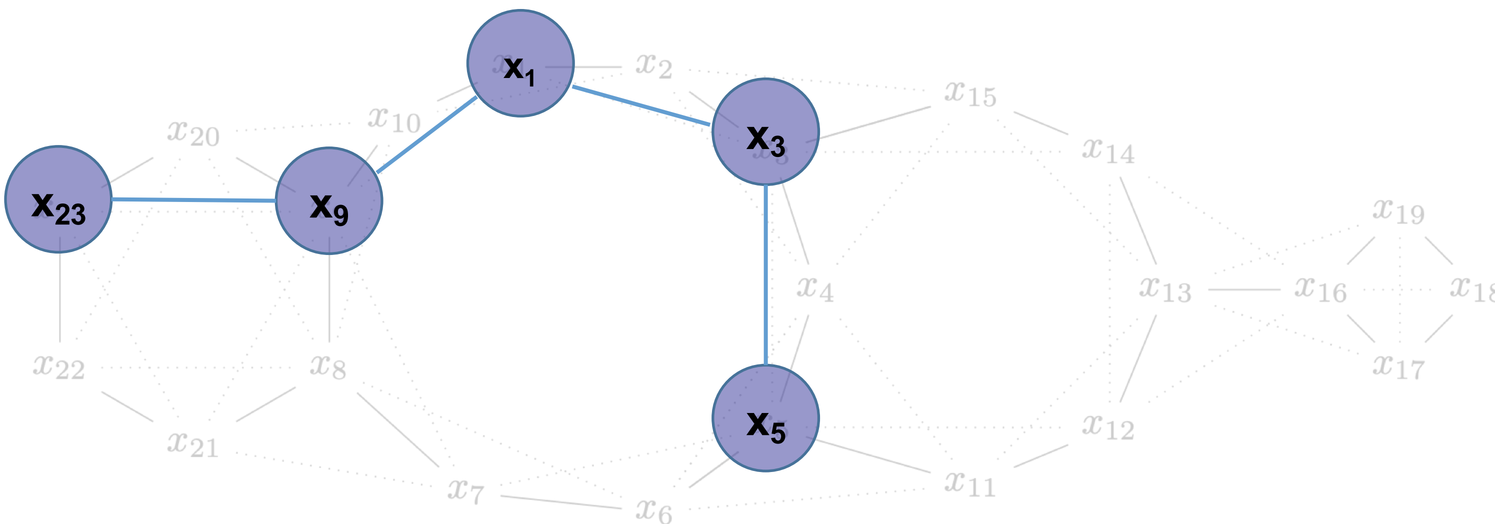


It requires time $O(2^n)$ in a black-box setting



It can be done in $O(4^\beta(n+m)+n^2)$ time in a gray-box setting for low-epistasis functions using DPX

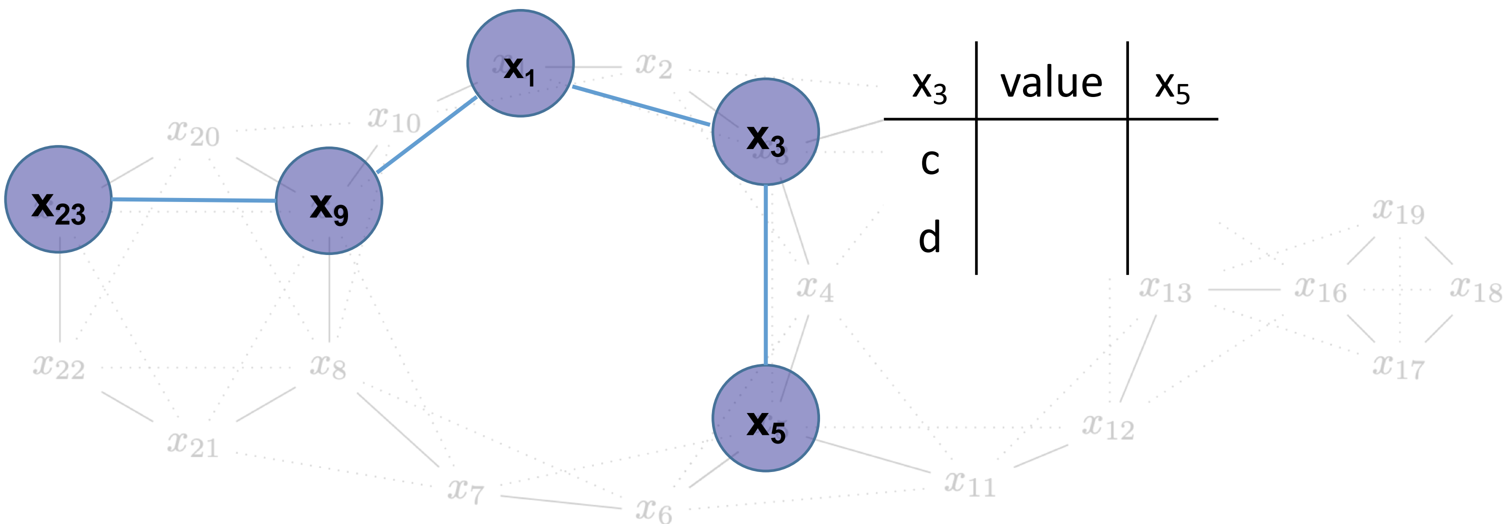
Dynastic Potential Crossover



We compute the optimum using dynamic programming by eliminating the variables

Hammer, Rosenberg, Rudeanu (1963)

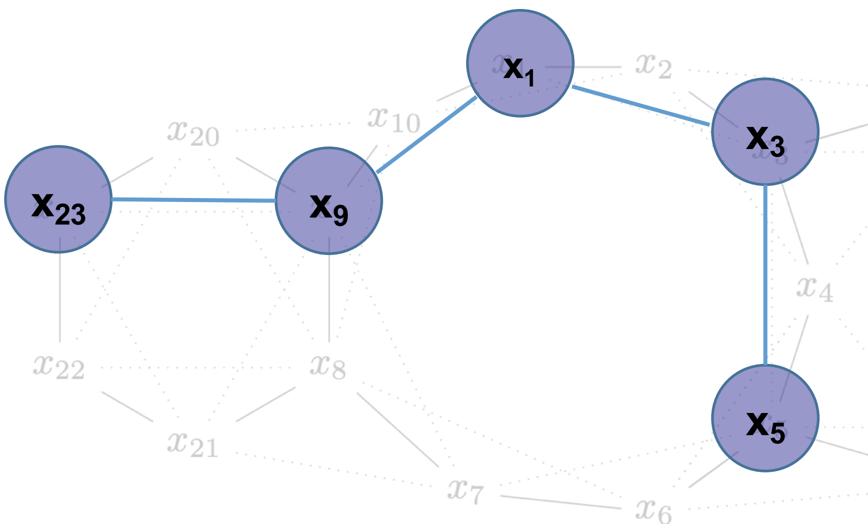
Dynastic Potential Crossover



x_3	value	x_5
c		
d		

We compute the optimum using dynamic programming by eliminating the variables

Dynastic Potential Crossover

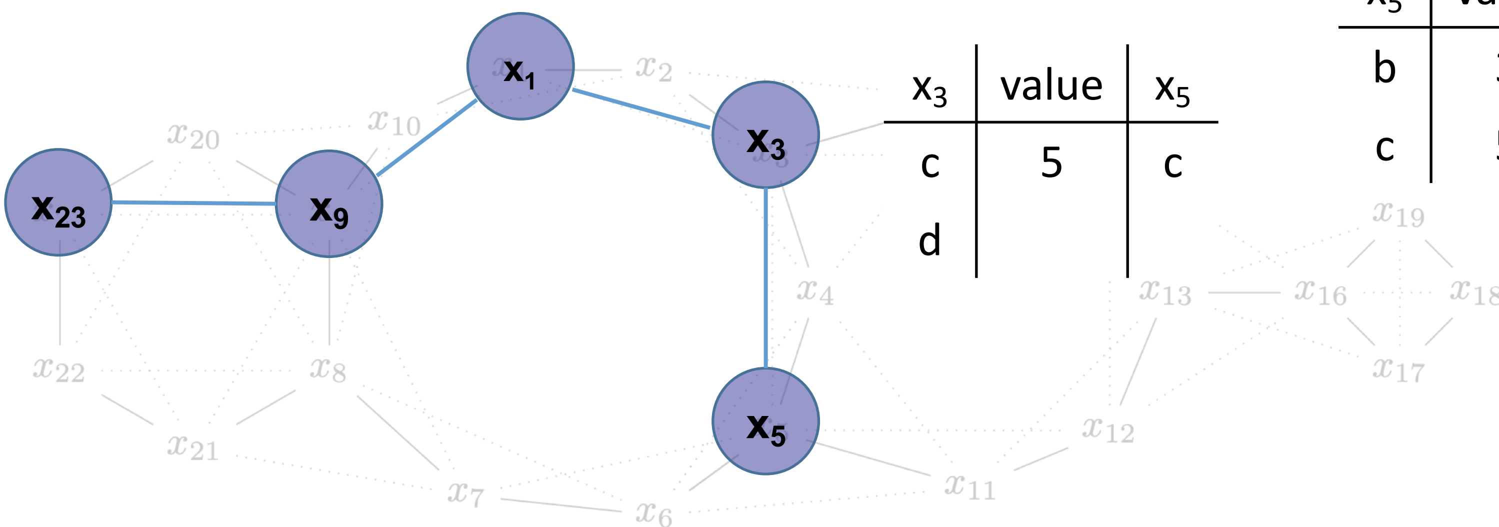


x_3	value	x_5
c		
d		

x_5	value
b	3
c	5

We compute the optimum using dynamic programming by eliminating the variables

Dynastic Potential Crossover

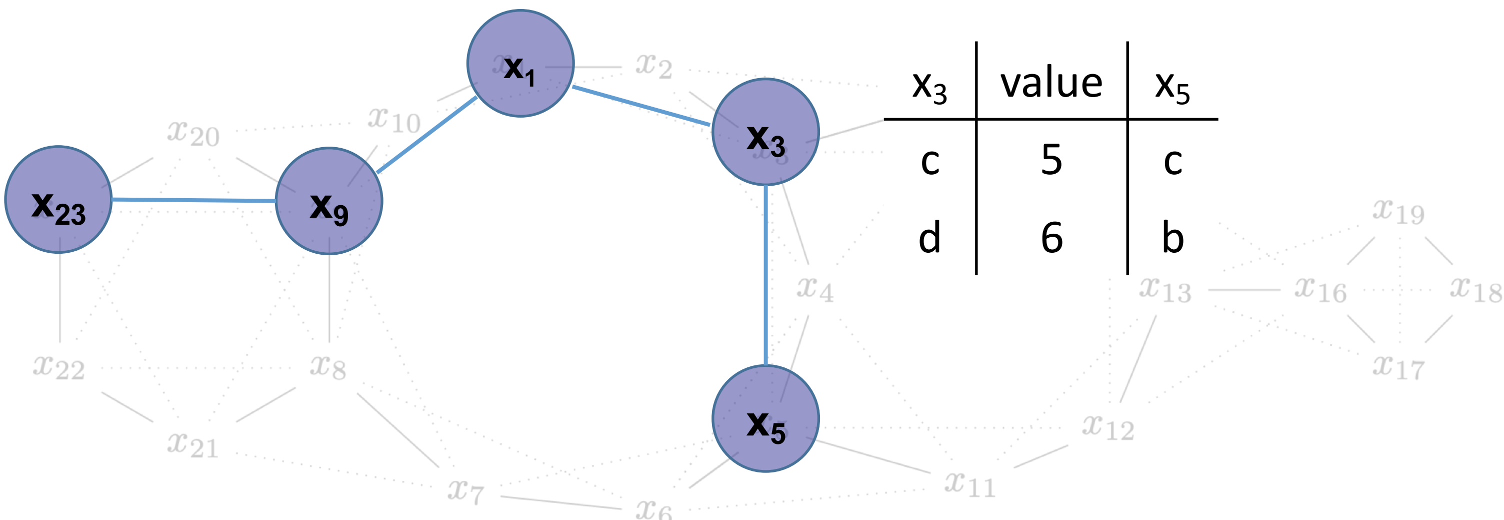


x_3	value	x_5
c	5	c
d		

x_5	value
b	3
c	5

We compute the optimum using dynamic programming by eliminating the variables

Dynastic Potential Crossover



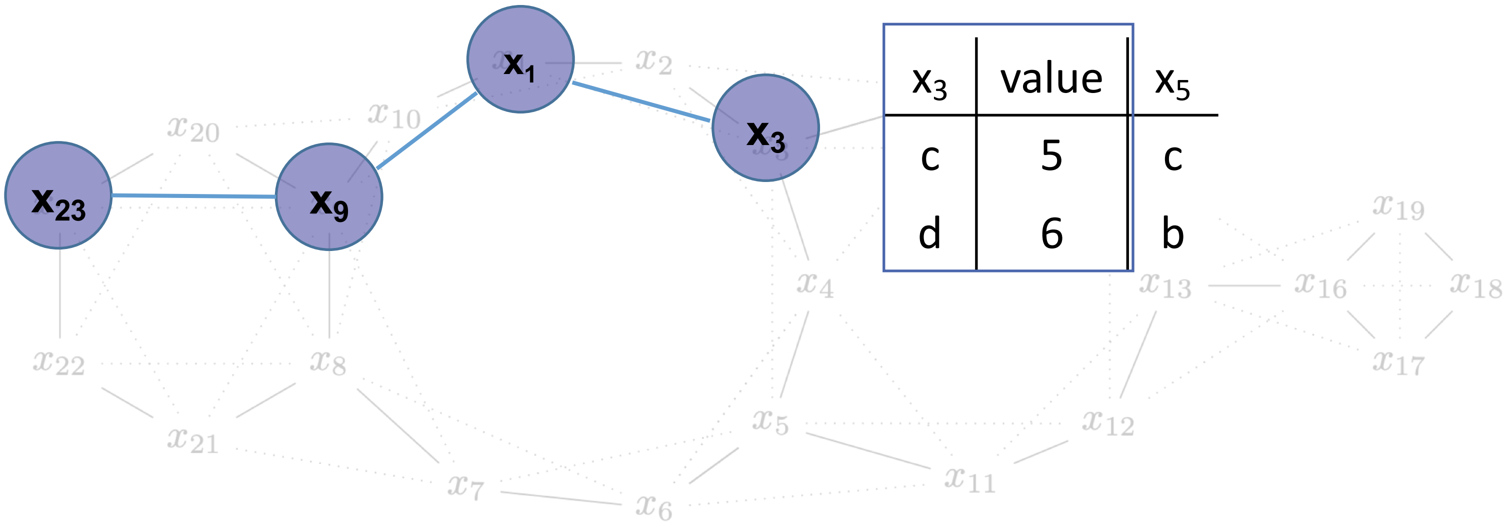
x_3	value	x_5
c	5	c
d	6	b

We compute the optimum using dynamic programming by eliminating the variables

Dynastic Potential Crossover

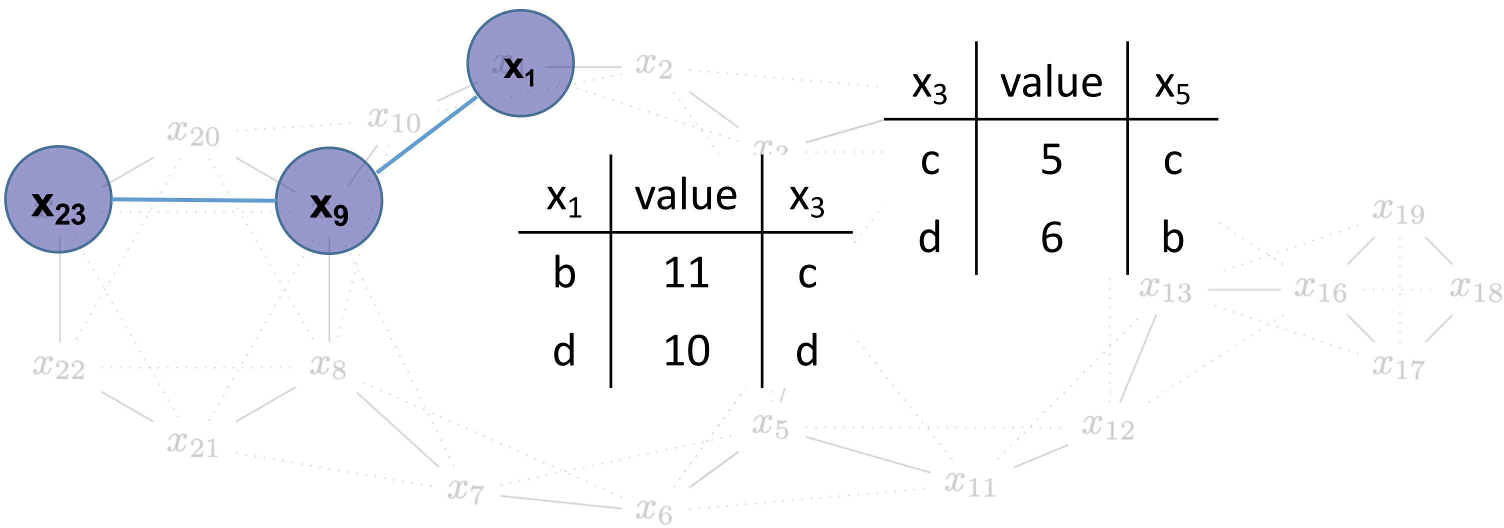
We add this subfunction

x_3	value	x_5
c	5	c
d	6	b



We compute the optimum using dynamic programming by eliminating the variables

Dynastic Potential Crossover



x_1	value	x_3
b	11	c
d	10	d

x_3	value	x_5
c	5	c
d	6	b

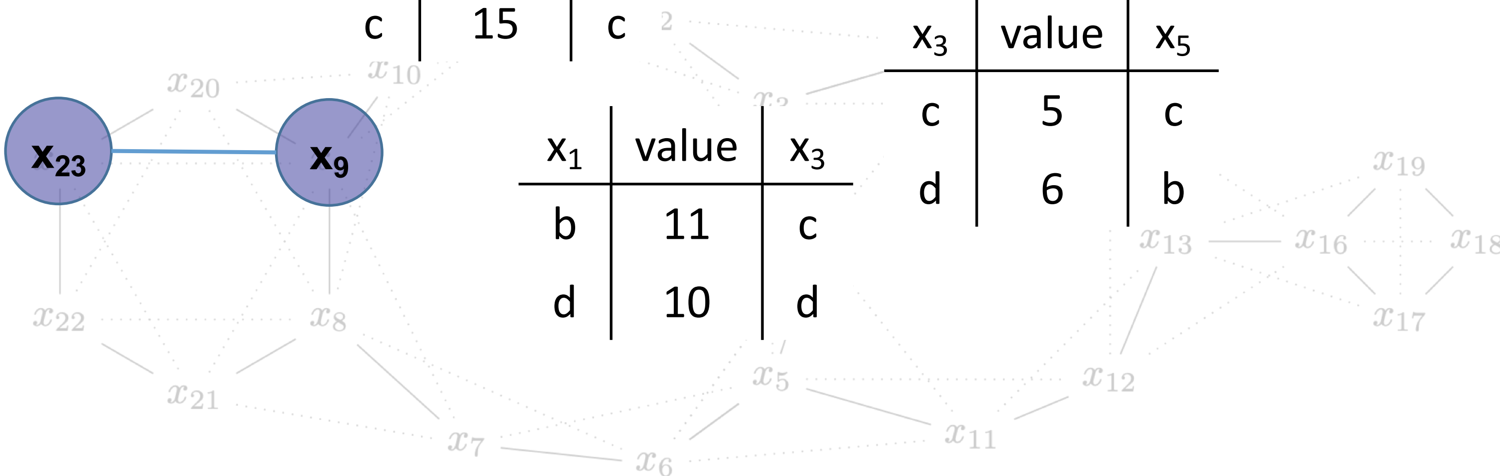
We compute the optimum using dynamic programming by eliminating the variables

Dynastic Potential Crossover

x_9	value	x_1
b	20	d
c	15	c

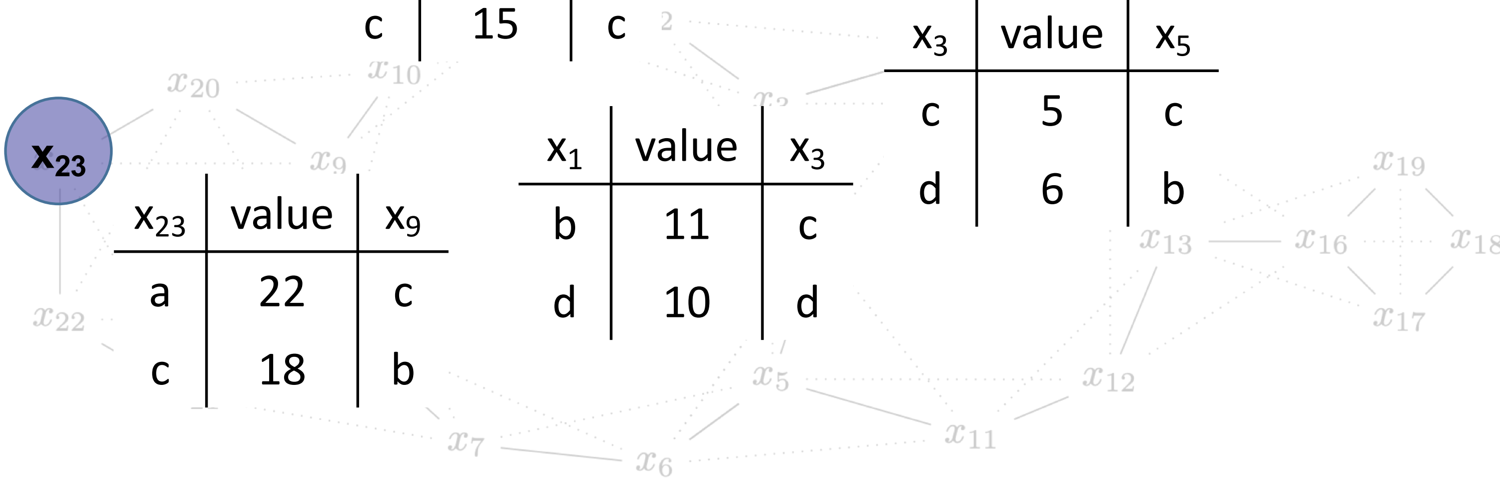
x_1	value	x_3
b	11	c
d	10	d

x_3	value	x_5
c	5	c
d	6	b



We compute the optimum using dynamic programming by eliminating the variables

Dynastic Potential Crossover



x_9	value	x_1
b	20	d
c	15	c

x_3	value	x_5
c	5	c
d	6	b

x_{23}	value	x_9
a	22	c
c	18	b

x_1	value	x_3
b	11	c
d	10	d

We compute the optimum using dynamic programming by eliminating the variables

SLS 2020 workshop
Held in conjunction with PPSN 2020

Dynastic Potential Crossover

x_9	value	x_1
b	20	d
c	15	d

x_3	value	x_5
c	5	c
d	6	b

x_1	value	x_3
b	11	c
d	10	d

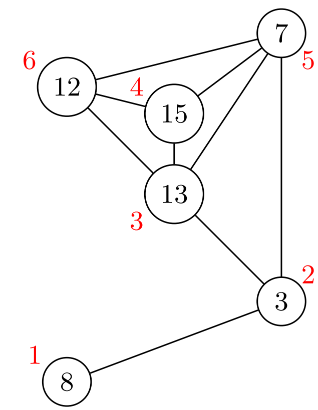
x_{23}

x_{23}	value	x_9
a	22	c
c	18	b

x_{23}	x_9	x_1	x_3	x_5
a	c	d	d	b

When one variable remains we take its best value and reconstruct the solution (optimal child)

Dynastic Potential Crossover



In general, we have to build a *clique tree* (*junction tree*)

Tarjan, Yannakakis (1984)

Galinier, Habib, Paul (1995)

Algorithm 1 Pseudocode of DPX

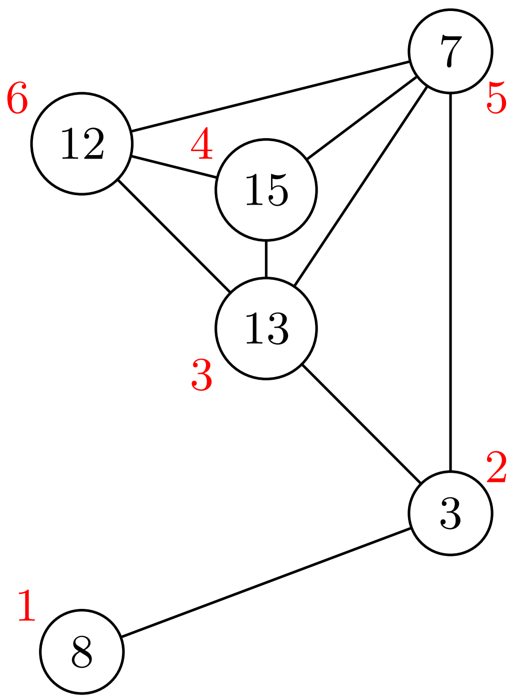
Input: two parents x and y
Output: one offspring z

Compute the Recombination Graph of x and y as in [6]
Apply Maximum Cardinality Search to the Recombination Graph [12]
Apply the fill-in procedure to make the graph chordal [12]
Apply the Clique Tree construction procedure [13]
Assign subfunctions to cliques in the clique tree
Apply Dynamic Programming to find the offspring (see Algorithm 2)
Build z using the tables filled by Dynamic Programming

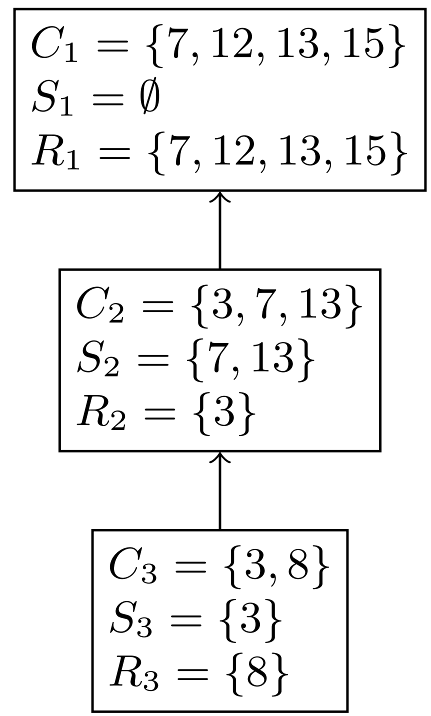
EvoCOP 2019: C., Ochoa, Whitley, Tinós

Dynastic Potential Crossover

Recombination graph

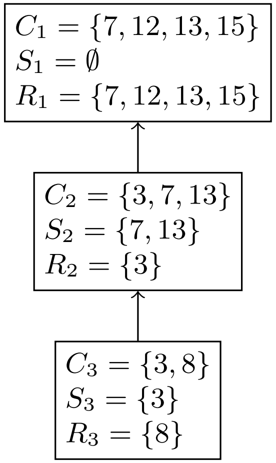


Clique tree



Dynastic Potential Crossover

What if we have too many variables to enumerate?



We put a limit β on the number fo variables that are fully enumerated

The remaining ones are taken in block from the parents

This makes the operator is Quasi-Optimal…

…and makes the time complextiy to be:

$$O(4^{\beta}(n+m)+n^2)$$

Experimental Results

Problems and Instances

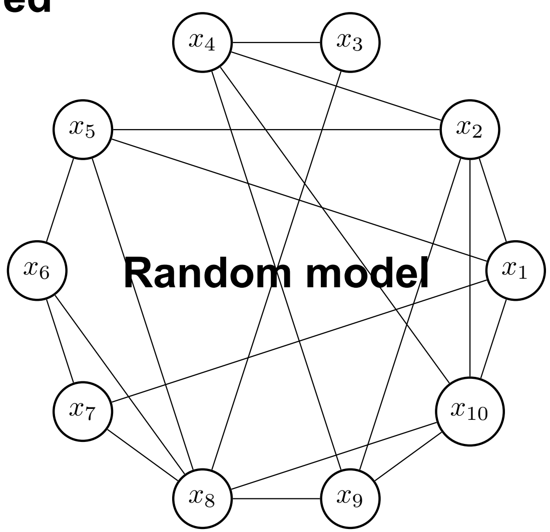
An NK Landscape is a pseudo-Boolean optimization problem with objective function:

$$f(x) = \sum_{l=1}^{N} f^{(l)}(x)$$

where each subfunction $f^{(l)}$ depends on variable x_l and K other variables

MAX-SAT consists in finding an assignment of variables to Boolean (true and false) values such that the maximum number of clauses is satisfied

A clause is an OR of literals: $x_1 \vee \neg x_2 \vee x_3$



SLS 2020 workshop
Held in conjunction with PPSN 2020

DPX Statistics with NKQ Landscapes

Table 1: Average runtime of crossover operators for random NKQ Landscapes with n =10 000 variables. Time is in microseconds (μs) for UX and in milliseconds (ms) for the rest. The Hamming distance between parents, h, is expressed in percentage of variables.

h	UX	NX	PX	APX	DPX (ms)					
%	μs	ms	ms	ms	$\beta = 0$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
					$K = 2$					
1	73	1.2	0.5	1.0	0.8	0.9	0.8	0.8	0.8	0.9
2	95	2.3	0.9	2.5	2.1	2.3	2.4	2.0	2.1	1.9
4	93	2.3	1.4	4.5	2.9	2.8	2.9	2.5	2.5	2.4
8	120	2.3	2.2	7.2	6.3	6.9	6.3	5.8	5.8	5.7
16	113	1.2	2.8	7.1	5.5	5.9	5.8	5.8	5.4	5.3
32	154	1.7	9.3	12.7	22.1	22.8	23.5	23.3	24.6	23.3
					$K = 5$					
1	68	3.2	0.9	2.6	1.4	1.5	1.5	1.4	1.4	1.3
2	82	3.7	1.8	5.2	2.1	2.3	2.3	2.1	2.2	2.0
4	85	4.2	3.5	8.7	3.6	3.9	3.8	3.9	4.0	4.1
8	119	4.3	5.4	13.3	8.0	8.1	8.2	9.5	10.9	9.9
16	113	3.0	4.1	12.8	90.7	83.0	103.0	92.2	101.3	107.5
32	139	3.7	5.8	19.4	1 000.5	1 034.0	1 041.1	1 020.3	1 089.9	1 021.7

SLS 2020 workshop
Held in conjunction with PPSN 2020

DPX Statistics with NKQ Landscapes

Table 2: Average quality improvement ratio of crossover operators for random NKQ Landscapes with n =10 000 variables. The numbers are in parts per thousand (‰). The Hamming distance between parents, h, is expressed in percentage of variables.

$$QIR_f(x,y,z) = \frac{f(z) - \max\{f(x), f(y)\}}{\max\{f(x), f(y)\}}$$

h	UX	NX	PX	APX	DPX (‰)					
%	‰	‰	‰	‰	$\beta = 0$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
					$K = 2$					
1	-0.58	-0.55	4.92	4.93	4.92	5.04	5.04	5.04	5.04	5.04
2	-0.79	-0.81	9.89	9.99	9.95	10.38	10.39	10.39	10.39	10.39
4	-1.13	-1.11	19.28	19.96	19.70	21.21	21.23	21.23	21.23	21.23
8	-1.56	-1.54	35.04	39.19	38.15	42.80	42.92	42.92	42.92	42.92
16	-2.08	-2.07	53.43	70.87	75.03	85.72	86.21	86.21	86.21	86.21
32	-2.72	-2.71	34.41	42.09	108.86	123.98	134.38	137.29	138.78	139.76
					$K = 5$					
1	-0.79	-0.78	6.38	6.72	6.61	7.18	7.18	7.18	7.18	7.18
2	-1.10	-1.10	11.46	13.40	13.17	14.77	14.81	14.81	14.81	14.81
4	-1.53	-1.56	15.06	20.38	26.44	29.58	30.06	30.14	30.16	30.17
8	-2.07	-2.06	8.07	9.56	31.18	34.54	39.26	41.02	41.98	42.67
16	-2.68	-2.66	2.19	2.90	30.14	31.61	37.08	41.51	43.48	44.83
32	-3.15	-3.13	0.28	0.77	32.42	32.82	34.18	36.64	40.31	44.05

DPX Statistics with NKQ Landscapes

Table 3: Average logarithm in base 2 of the solutions explored by PX, APX and DPX for random NKQ Landscapes with n =10 000 variables. The Hamming distance between parents, h, is expressed in percentage of variables.

h	PX	APX	DPX ($\log_2$)					
%	$\log_2$	$\log_2$	$\beta=0$	$\beta=1$	$\beta=2$	$\beta=3$	$\beta=4$	$\beta=5$
			$K=2$					
1	97.1	97.3	97.2	100.0	100.0	100.0	100.0	100.0
2	188.1	190.3	189.3	199.9	200.0	200.0	200.0	200.0
4	352.9	368.1	362.0	399.5	400.0	400.0	400.0	400.0
8	613.5	703.3	679.7	796.8	800.0	800.0	800.0	800.0
16	873.6	1 220.6	1 311.2	1 586.5	1 600.0	1 600.0	1 600.0	1 600.0
32	660.7	828.3	2 055.6	2 399.2	2 586.9	2 636.5	2 661.3	2 677.4
			$K=5$					
1	85.4	91.1	89.1	99.9	100.0	100.0	100.0	100.0
2	142.0	172.4	168.5	199.2	200.0			
4	175.4	246.1	332.2	390.6	398.2			
8	113.2	132.7	420.5	470.8	530.8			
16	38.9	47.6	449.0	469.0	542.8	601.9	627.7	645.3
32	7.5	13.7	534.0	539.3	559.3	595.7	649.7	703.5

Full dynastic potential (2^{1600}) explored in 5.3 ms

Experiments in Search Algorithms

Two Algorithms were used:

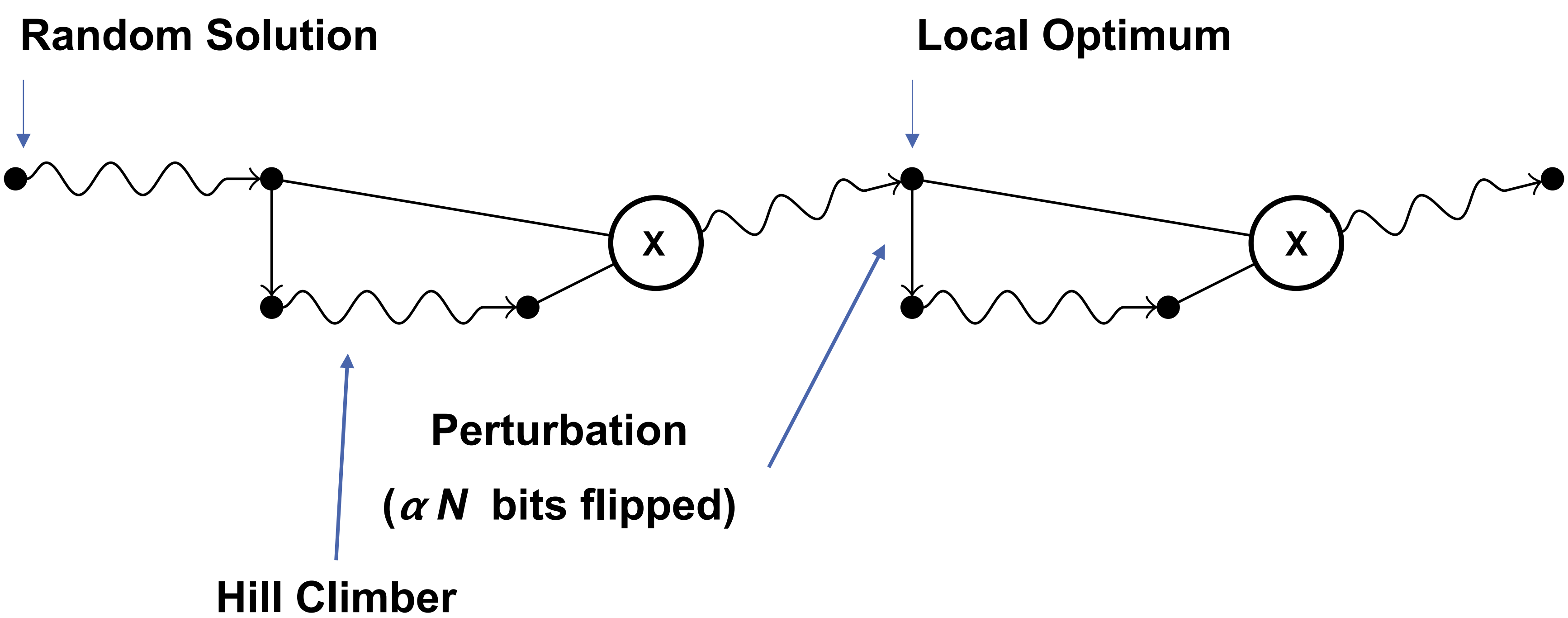
- **Steady-state Evolutionary Algorithm (population-based metaheuristic)**
- **DRILS (trajectory-based metaheuristic)**

The parameters of the algorithms were tuned using irace for each class of instances

- **NKQ Landscapes with K=2 and K=5**
- **MAX-SAT (industrial and crafted) instances from MAX-SAT Evaluation 2017**
  - **160 unweighted**
  - **132 weighted**

SLS 2020 workshop
Held in conjunction with PPSN 2020

Deterministic Recombination and Iterated Local Search (DRILS)



Included in DRILS and EA in NKQ Landscapes

Table 8: Performance of the five recombination operators used in DRILS and EA when solving NKQ Landscapes instances with $n = 10\,000$ variables. The symbols ▲, ▽ and = are used to indicate that the use of the crossover operator in the row yields statistically better, worse or similar results than the use of DPX in each algorithm.

	$K = 2$		$K = 5$	
	Statistical difference	Quality	Statistical difference	Quality
DRILS				
DPX		0.9997		0.9972
APX	0▲ 8▽ 2 =	0.9995	0▲ 7▽ 3 =	0.9947
PX	0▲ 10▽ 0 =	0.9990	0▲ 7▽ 3 =	0.9949
NX	0▲ 10▽ 0 =	0.9786	0▲ 10▽ 0 =	0.9934
UX	0▲ 10▽ 0 =	0.9790	0▲ 10▽ 0 =	0.9935
EA				
DPX		0.9795		0.8132
APX	0▲ 10▽ 0 =	0.9568	1▲ 0▽ 9 =	0.8890
PX	0▲ 10▽ 0 =	0.9445	10▲ 0▽ 0 =	0.9085
NX	0▲ 10▽ 0 =	0.8803	0▲ 1▽ 9 =	0.7811
UX	0▲ 10▽ 0 =	0.9313	0▲ 1▽ 9 =	0.8407

SLS 2020 workshop
Held in conjunction with PPSN 2020

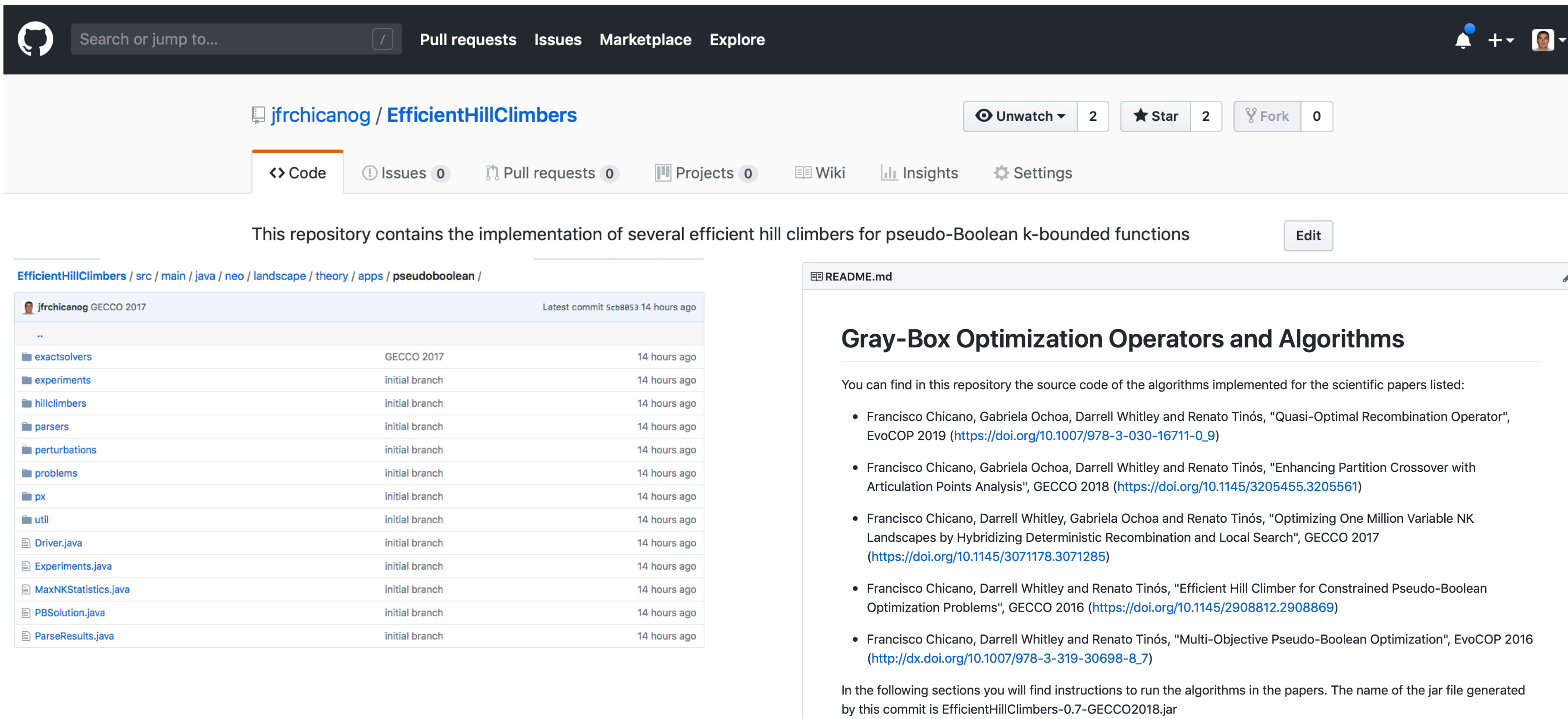
Included in DRILS and EA in MAX-SAT

Table 9: Performance of the five recombination operators used in DRILS and EA when solving MAX-SAT instances. The symbols ▲, ▽ and = are used to indicate that the use of the crossover operator in the row yields statistically better, worse or similar results than the use of DPX.

	Unweighted		Weighted	
	Statistical difference	Quality	Statistical difference	Quality
DRILS				
DPX		0.9984		0.9996
APX	14▲ 91▽ 57 =	0.9973	15▲ 86▽ 31 =	0.9984
PX	8▲ 103▽ 55 =	0.9968	25▲ 80▽ 27 =	0.9982
NX	2▲ 126▽ 28 =	0.9946	1▲ 126▽ 5 =	0.9915
UX	0▲ 124▽ 40 =	0.9953	1▲ 126▽ 5 =	0.9930
EA				
DPX		0.9644		0.9583
APX	52▲ 68▽ 40 =	0.9604	43▲ 63▽ 26 =	0.9649
PX	17▲ 107▽ 36 =	0.9095	8▲ 109▽ 15 =	0.9057
NX	18▲ 101▽ 41 =	0.8980	18▲ 103▽ 11 =	0.8786
UX	27▲ 96▽ 37 =	0.9134	18▲ 99▽ 15 =	0.8989

Source Code in GitHub

https://github.com/jfrchicanog/EfficientHillClimbers



Conclusions

- **DPX is a very effective crossover operator**
- **Main drawback: runtime and memory consumption**
- **"Removing randomness" from metaheuristic algorithms (D. Whitley)**
- **Take home message: use Gray-Box Optimization if you can**

Future Work

- **Explore the shape of the connected components in the recombination graph and their relationship with performance**
- **Find the optimal value of the parameters using the VIG**

References (I)

- **Chen, W., Whitley, D., 2017. Decomposing SAT Instances with Pseudo Backbones. EvoCOP: 75-90**
- **Chicano, F., Ochoa, G., Whitley, D., Tinós, R., 2018. Enhancing partition crossover with articulation points analysis. GECCO: 269-276**
- **Chicano, F., Ochoa, G., Whitley, D., Tinós, R., 2019. Quasi-Optimal Recombination Operator. EvoCOP: 131-146**
- **Eremeev, A. V., Kovalenko, Y. V., 2013. Optimal Recombination in Genetic Algorithms. CoRR abs/1307.5519**
- **Galinier, P., Habib, M., Paul, C., 1995. Chordal graphs and their clique graphs. In Graph-Theoretic Concepts in Computer Sciences: 358-371**
- **Hammer, P. L., Rosenberg, I., Rudeanu, S., 1963. On the determination of the minima of pseudo-boolean functions. Stud. Cerc. Mat. 14:359-364**
- **Radcliffe, N. J., 1994. The algebra of genetic algorithms. Annals of Mathematics and Artificial Intelligence 10(4): 339-384**

SLS 2020 workshop
Held in conjunction with PPSN 2020

References (II)

- **Tinós, R., Whitley, D., Chicano, F., 2015. Partition Crossover for Pseudo-Boolean Optimization. FOGA: 137-149**
- **Tarjan, R. E., Yannakakis, M., 1984. Simple linear-time algorithms to test chordality of graphs, test acyclicity and hypergraphs, and selectively reduce acylic hypergraphs. SIAM Journal on Computing 13(3): 566-579**

Acknowledgements

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD

UNIVERSITAS MALACITANA
UNIVERSIDAD DE MÁLAGA

ANDALUCÍA TECH
Campus de Excelencia Internacional

Thanks for your attention!!!