Improving Local Search Algorithms via Probabilistic Configuration Checking

Weilin Luo[1], Rongzhen Ye[1], Hai Wan[1,*], Shaowei Cai[2,*], Biqing Fang[1], Delong Zhang[1]
[1]Sun Yat-sen University, Guangzhou, China [2]Chinese Academy of Sciences, China

AAAI-22

Motivation

Local search is a popular method for solving combinatorial optimization problems (COPs), *e.g.*

- **minimum weight dominating set problem (MWDS)**
- **maximum weight clique problem (MWCP)**

Configuration checking (CC) is an effective mechanism to alleviate the *cycling problem* for COPs.

The configuration of a vertex in the previous CC strategies:

- MWDS: 2-level neighborhoods
- MWCP: 1-level neighborhoods

Neighborhoods with more than two levels can have an impact on a vertex and should be considered in the configuration.

Instances	CC²FS			CC²FS-H		
	min	avg	time	min	avg	time (s)
frb30-15-1	212	214.2	2.59	212	**212.0**	**0.11**
frb30-15-2	242	242.0	8.81	242	242.0	**0.18**
frb30-15-3	175	175.0	**0.01**	175	175.0	0.03
frb30-15-4	166	168.6	0.19	166	**166.0**	**0.08**
frb30-15-5	160	160.0	**0.01**	160	160.0	0.02
frb35-17-1	274	274.9	0.57	274	**274.0**	**0.11**
frb35-17-2	208	208.3	67.44	208	**208.0**	**0.29**
frb35-17-3	201	201.0	0.38	201	201.0	**0.10**
frb35-17-4	286	286.8	0.73	286	**286.0**	**0.45**
frb35-17-5	295	295.7	5.60	295	**295.0**	**0.11**
frb40-19-1	262	262.0	1.99	262	262.0	**0.14**
frb40-19-2	243	243.9	0.66	243	**243.0**	**0.18**
frb40-19-3	250	251.8	102.43	250	**250.0**	**3.88**
frb40-19-4	250	250.0	0.07	**249**	**249.0**	95.82
frb40-19-5	272	282.3	0.31	272	**277.6**	82.50
frb45-21-1	328	328.3	7.82	328	**328.0**	**0.64**
frb45-21-2	259	261.7	46.82	259	**259.1**	**1.39**
frb45-21-3	233	233.9	9.19	233	**233.0**	**0.89**
frb45-21-4	399	399.2	22.67	399	**399.0**	**0.95**
frb45-21-5	312	318.8	128.19	312	**312.0**	**1.82**
frb50-23-1	261	264.7	90.71	261	**261.0**	**8.97**
frb50-23-2	277	277.0	**0.01**	277	277.0	0.17
frb50-23-3	299	301.9	1.21	**281**	**292.2**	31.13
frb50-23-4	265	265.0	3.04	265	265.0	**0.47**
frb50-23-5	410	418.8	128.09	410	**411.1**	158.96
frb53-24-1	229	230.0	95.41	229	**229.0**	**2.35**
frb53-24-2	298	298.0	1.04	298	298.0	**0.38**
frb53-24-3	182	182.1	0.01	182	**182.0**	0.26
frb53-24-4	189	189.0	**0.02**	189	189.0	0.27
frb53-24-5	204	204.0	**0.01**	204	204.0	0.29
frb56-25-1	229	229.2	0.32	229	**229.0**	0.41
frb56-25-2	319	319.5	25.27	319	**319.0**	**3.46**
frb56-25-3	336	338.7	30.21	336	**336.0**	38.80
frb56-25-4	268	268.0	**0.21**	268	268.0	0.41
frb56-25-5	425	427.6	103.13	425	**425.0**	**1.83**
frb59-26-1	262	264.0	1.62	262	**262.0**	25.31
frb59-26-2	383	391.1	97.04	383	**383.0**	**5.64**
frb59-26-3	248	248.0	1.13	**246**	**246.0**	14.24
frb59-26-4	248	248.9	47.47	248	**248.0**	**0.62**
frb59-26-5	288	288.8	195.62	288	**288.0**	**3.26**
frb100-40	350	350.0	**2.04**	350	350.0	9.70
#win	0	0	7	**3**	**29**	**25**

Contributions

1. **Probabilistic configuration (PC)**. A general way to define the configuration of a vertex.
2. **Probabilistic configuration checking (PCC)**. The strategy of the new configuration.

i-th-level Neighborhoods

Definition 1 *For a vertex v, we define its i-th-level neighborhoods as $N_i(v) = \{u \in V \mid dist(u,v) = i\}$, i-level neighborhoods as $N_{\leq i}(v) = \bigcup_{j=1}^{i} N_j(v)$, and its i^+-level neighborhoods as $N_{>i}(v) = \bigcup_{j=i+1} N_j(v)$, particularly, $N_1(v) = N_{\leq 1}(v) = N(v)$.*

Probabilistic Configuration (PC)

Definition 2 *Given an undirected graph $G = (V, E)$ and a candidate solution S, the probabilistic configuration (PC) of a vertex $v \in V$ is a vector consisting of the states of all vertices in $(\bigcup_{i=1}^{2}(N_i(v))^{p_i}) \cup (N_{>2}(v))^{p_3}$, where $p_i \in [0,1]$, $i = 1,2,3$, and $\{*\}^{p_i}$ is randomly chosen vertices from the set $\{*\}$ with the size of $\lceil |\{*\}| \cdot p_i \rceil$.*

Probabilistic Configuration Checking (PCC)

Algorithm 1 SelectConf(v, $N_1(v)$, $N_2(v)$, $N_{>2}(v)$, P)

Input: a vertex v and $N_1(v)$, $N_2(v)$, $N_{>2}(v)$, $P = \{p_1, p_2, p_3\}$ a set of probabilities.
Output: the configuration $Conf$.

1: $Q_1 \leftarrow$ randomly choose $\lceil |N_1(v)| \cdot p_1 \rceil$ vertices from $N_1(v)$;
2: $Q_2 \leftarrow$ randomly choose $\lceil |N_2(v)| \cdot p_2 \rceil$ vertices from $N_2(v)$;
3: $Q_3 \leftarrow$ randomly choose $\lceil |N_{>2}(v)| \cdot p_3 \rceil$ vertices from $N_{>2}(v)$;
4: $Conf(v) \leftarrow Q_1 \cup Q_2 \cup Q_3$;
5: **return** $Conf(v)$;

- Firstly, carry out a preprocessing to compute the 1st-, 2nd-, and 2^+- level neighborhoods for each vertex, which can be quickly implemented with a breadth-first search algorithm with the time complexity of $O(|V| \cdot |E|)$, which only needs to be run once.
- In each search step, for the operated vertex v, select vertices from $N_1(v)$, $N_2(v)$ and $N_{>2}(v)$ to form its configuration (Algorithm 1).

Comparison PCC with CC

MWDS: CC²FS, CC²FS-P (ours, one PC for each COP)

	Benchmarks	#inst.	CC²FS #w.m.	CC²FS #w.a.	CC²FS-P #w.m.	CC²FS-P #w.a.	sum of diff min	sum of diff avg
train	T1	5	0	0	**5**	**5**	**-167**	**-167.0**
	T2	5	0	0	**3**	**3**	**-49**	**-49.0**
	UDG	4	**4**	**4**	0	0	30	30.0
	BHOSLIB	4	0	0	**1**	**4**	**-1**	**-8.4**
	DIMACS	8	0	0	0	0	0	0.0
test	T1	525	10	10	**152**	**152**	**-2129**	**-2129.0**
	T2	525	0	0	**57**	**57**	**-697**	**-697.0**
	UDG	116	**81**	**81**	16	16	541	541.0
	BHOSLIB	37	0	0	**6**	**26**	**-52**	**-105.1**
	DIMACS	29	0	0	**2**	**9**	**-4**	**-27.2**
Total		1258	95	95	**242**	**272**	**-2528**	**-2611.7**

MWCP: LSCC, LSCC-P (ours, one PC for each COP)

	Benchmarks	#inst.	LSCC #w.m.	LSCC #w.a.	LSCC-P #w.m.	LSCC-P #w.a.	sum of diff min	sum of diff avg
train	BHOSLIB	4	1	0	**2**	**4**	**13**	**67**
	DIMACS	8	0	1	**3**	**3**	**124**	**166.8**
test	BHOSLIB	37	2	3	**14**	**26**	**315**	**462.7**
	DIMACS	29	**2**	2	0	2	-19	**52.9**
Total		78	5	6	**19**	**35**	**433**	**749.4**

Discussion About the Bias of Training

CC²FS-P performs worse even in the training set with respect to the UDG benchmark in MWDS. Therefore, we argue that it results from the adverse bias generated from training data from other benchmarks.

The results below show that the PC specially trained for each benchmark can improve the performance.

Discussion About the Bias of Training

MWDS: CC²FS-P, CC²FS-SP (ours, one PC for each benchmark)

	Benchmarks	#inst.	CC²FS-P #w.m.	CC²FS-P #w.a.	CC²FS-SP #w.m.	CC²FS-SP #w.a.	sum of diff min	sum of diff avg
train	T1	5	0	0	**2**	**2**	**-17**	**-17.0**
	T2	5	0	0	0	0	0	0.0
	UDG	4	0	0	**4**	**4**	**-38**	**-38.0**
	BHOSLIB	4	0	0	0	0	0	0.0
	DIMACS	8	0	0	0	0	0	0.0
test	T1	525	17	17	**36**	**36**	**-76**	**-76.0**
	T2	525	1	1	1	1	0	0.0
	UDG	116	22	22	**82**	**82**	**-555**	**-555.0**
	BHOSLIB	37	0	0	0	**1**	0	**-4.0**
	DIMACS	29	0	**1**	0	0	0	0.1
Total		1258	40	41	**125**	**126**	**-686**	**-689.9**

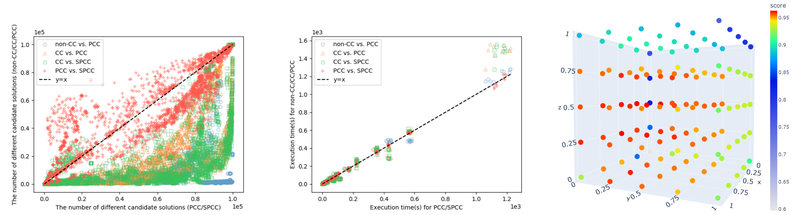
MWCP: LSCC-P, LSCC-SP (ours, one PC for each benchmark)

	Benchmarks	#inst.	LSCC-P #w.m.	LSCC-P #w.a.	LSCC-SP #w.m.	LSCC-SP #w.a.	sum of diff max	sum of diff avg
train	BHOSLIB	4	1	2	**2**	2	-5	**2.4**
	DIMACS	8	0	0	0	0	0	0
test	BHOSLIB	37	**9**	8	6	**15**	**56**	**39.6**
	DIMACS	29	0	0	0	0	0	0
Total		78	**10**	10	8	**17**	**51**	**42**

Reducing the Cycling Problem

first 10^5 candidate solutions.

(a) Comparison of the number of different candidate solutions of non-CC, CC, PCC, SPCC.
(b) Comparison of CPU time.
(c) The scores of CC²FS-P with different configurations on MWDS benchmark. The X-axis, Y-axis, and Z-axis represent p_1, p_2, and p_3 respectively.



- The result shows that PCC does not have much more time consumption to improve the exploration in local search and potentially alleviates the cycling problem.
- It is necessary to tune for an optimal PC for a given COP.

Acknowledgments

We thank anonymous referees for helpful comments. This paper was supported by the National Natural Science Foundation of China (No. 61976232), Humanities and Social Science Research Project of Ministry of Education (18YJCZH006).