Supplementary Materials: A Novel Local Search Algorithm for the Vertex Bisection Minimization Problem

1 Deep Analysis of the Two-Manner Based Local Search

In this section, we present a deep analysis of the two-manner-based local search. Specifically, we evaluate a total of nine combinations of swap operations. First, we present the execution condition for the swap operation (Line 5 in Algorithm 5) in CELS and another two different execution conditions for the swap operation as follows:

- $|VM(D)| \leq VM(D_{\text{best}})$,
- $|VM(D)| \leq VM(D_{\text{best}}) + 1$,
- $|VM(D)| \leq VM(D_{\text{best}}) + 2$.

Additionally, we present the swap rule utilized by CELS and another two alternate swap rules , where the swap rules become gradually greedy:

- **Swap Rule:** Vertices with the lowest $\Delta_{\text{drop}}(D, v)$ from $B \setminus V_{\text{lock}}$ are stored in S_B, and vertices with the lowest $\Delta_{\text{add}}(D, v)$ from $B' \setminus V_{\text{lock}}$ are stored in $S_{B'}$. If either set contains more than K vertices, we randomly retain K vertices in the corresponding set, where K is a predefined parameter. Then, we select a pair of vertices $(v, u) := \arg\min_{v \in S_B, u \in S_{B'}} \Delta_{\text{swap}}(D, v, u)$, breaking ties randomly.
- **Swap Rule 1:** Vertices with the lowest $\Delta_{\text{drop}}(D, v)$ from $B \setminus V_{\text{lock}}$ are stored in S_B, and vertices with the lowest $\Delta_{\text{add}}(D, v)$ from $B' \setminus V_{\text{lock}}$ are stored in $S_{B'}$. Then, we directly select a pair of vertices $(v, u) := \arg\min_{v \in S_B, u \in S_{B'}} \Delta_{\text{swap}}(D, v, u)$, breaking ties randomly.
- **Swap Rule 2:** A pair of vertices is selected as $(v, u) := \arg\min_{v \in B \setminus V_{lock}, u \in B' \setminus V_{lock}} \Delta_{\text{swap}}(D, v, u)$, breaking ties randomly.

By combining the three execution conditions with the three swap rules, we generate nine types of swap operations, one of which is adopted by CELS. We then implement the other eight modified versions of the swap operation in CELS under the same settings, and compare the best solutions obtained by CELS with those produced by each corresponding algorithm. The results are summarized in Table 1. From Table 1, it is clear that the swap operation used in CELS outperforms the other swap operations, indicating the effective design of the swap operation in CELS.

2 Detailed Results of All Algorithms

We present the results of CELS and the comparative algorithms in Tables 2–5. For each instance, min represents the smallest size achieved by each algorithm, while avg indicates the average size obtained across 10 runs. If min equals avg, the average value is omitted. The Pre_best column refers to the best results previously reported in the literature. In the tables, bold values highlight the best solutions achieved among all algorithms.

References

Swap Rule \ Execution Condition	$\lvert VM(D)\rvert \leq VM(D_{\text{best}})$	$\lvert VM(D)\rvert \leq VM(D_{\text{best}})+1$	$\lvert VM(D)\rvert \leq VM(D_{\text{best}})+2$
Swap Rule	*	**29** (10)	**42**(17)
Swap Rule 1	**19** (10)	**24** (13)	**39** (5)
Swap Rule 2	**23** (11)	**25** (12)	**47** (1)

Table 1: Comparison of CELS with eight modified versions. Each cell contains two numbers: the first represents the number of instances where CELS achieves a better minimal solution, and the second indicates the number of instances where it obtain a worse minimal solution.

Instance	Pre_best	CELS max(avg)	BVNSBucket max(avg)	BVNSBucket2 max(avg)	CLHUS max(avg)	BVNS max(avg)
g1	398	**384(384.9)**	399(399.7)	399(399.6)	387	399(399.3)
g2	399	**387(387.9)**	399(399.5)	398(399.8)	388	398(399.4)
g3	399	**384(384.9)**	399(399.6)	399(399.8)	386	398(399.2)
g4	398	387(388.2)	399(399.7)	399(399.6)	**385**	398(399.4)
g5	398	**384(384.8)**	399(399.8)	399(399.5)	386	399(399.7)
g6	398	**384(384.9)**	399(399.8)	399(399.6)	387	399(399.5)
g7	398	**387(387.9)**	399(399.5)	398(399.8)	388	399(399.5)
g8	399	**384(384.9)**	399(399.6)	399(399.8)	386	399(399.7)
g9	398	387(388.2)	399(399.7)	399(399.6)	**385**	399(399.6)
g10	398	**384(384.8)**	399(399.8)	399(399.5)	386	399(399.6)
g11	16	**16**	**16**	**16**	**16**	**16**
g12	32	**32**	**32**	**32**	**32**	**32**
g13	50	**50**	**50**	**50**	**50**	**50**
g14	188	**184(186)**	187(190.4)	187(188.1)	185(186.3)	188(192)
g15	184	**183**(184.7)	185(186.4)	185(186.2)	**183(184)**	185(188.4)
g16	188	**185(185.7)**	188(190)	188(189.9)	**185(185.7)**	188(189.8)
g17	182	**180(181.7)**	182(183.4)	182(183.7)	181(181.8)	181(184.9)
g18	187	**184(186)**	187(190.4)	187(188.1)	185(186.3)	189(192)
g19	184	**183**(184.7)	185(186.4)	185(186.2)	**183(184)**	186(187.7)
g20	187	**185(185.7)**	188(190)	188(189.9)	**185(185.7)**	187(189.8)
g21	180	**180(181.7)**	182(183.4)	182(183.7)	181(181.8)	182(184.2)
g22	875	**860(861.6)**	873(874.8)	868(873.4)	862(864.5)	889(895.2)
g23	877	**863(864.8)**	874(877.9)	874(876.8)	866(868.9)	890(894.2)
g24	874	**861(862.4)**	874(876.4)	873(875.5)	865(866.4)	880(890.3)
g25	876	**861(863.7)**	875(877.1)	873(875.5)	865(868.5)	892(899.8)
g26	877	**860(862.2)**	875(876.2)	873(876)	864(866.8)	885(897)
g27	875	**860(861.6)**	873(874.8)	868(873.4)	862(864.6)	879(889.6)
g28	878	**863(866)**	877(879.6)	876(878)	864(867)	896(900.7)
g29	874	**861(862.4)**	874(876.4)	873(875.5)	865(866.4)	883(894.4)
g30	876	**861(863.7)**	875(877.1)	873(875.5)	865(868.5)	890(894.8)
g31	877	**860(862.2)**	875(876.2)	873(876)	864(866.7)	887(893.7)
g32	40	**40**	**40**	**40**	**40**	**40**
g33	50	**50**	**50**	**50**	**50**	**50**
g34	80	**80**	**80**	**80**	**80**	**80**
g35	470	**454(456.3)**	465(471.1)	466(472)	456(458.1)	475(485.4)
g36	471	**457(462)**	470(477.4)	464(474.6)	459(462.5)	480(485.7)
g37	463	**453(455.9)**	463(471.9)	463(469.1)	456(458.2)	478(489.8)
g38	474	**455(458.4)**	465(473.4)	465(472.9)	457(461.9)	474(482.5)
g39	470	**454(456.3)**	465(471.1)	466(472)	456(458.1)	478(488.5)
g40	471	**457(462)**	470(477.4)	464(474.6)	459(462.5)	475(485.1)
g41	463	**453(455.9)**	463(471.9)	463(469.1)	456(458.2)	478(487.5)
g42	474	**455(458.4)**	465(473.4)	465(472.9)	457(461.9)	468(484.5)
g43	437	**430(431.1)**	436(437.9)	436(437.1)	**430(431.1)**	436(438)
g44	437	**430**(431.2)	437(437.9)	437(437.5)	**430(431.1)**	437(439.6)
g45	434	**429(430)**	435(436.6)	434(436.2)	**429(430)**	437(437.9)
g46	437	431**(431.4)**	436(438.2)	436(438)	**430**(431.8)	438(439.7)
g47	438	**430**(431.6)	436(438.7)	437(438.2)	**430(431.5)**	438(438.9)
g48	100	120	**100**	**100**	120	**100**
g49	60	**60**	**60**	**60**	**60**	**60**
g50	50	**50**	**50**(55)	**50**	**50**	**50**(55)
g51	234	**233**(233.7)	235(237)	**232**(236.1)	234(234.6)	235(238.3)
g52	233	**231**(232.2)	**230**(236.7)	232(236.4)	233(233.9)	236(237.7)
g53	232	**229(230.1)**	231(234.5)	230(234.7)	230(231.5)	234(236.6)
g54	229	**228(228.4)**	229(231.2)	**228**(230.8)	**228**(229.3)	229(232.6)
g55	1052	**1014(1023)**	1047(1065.5)	1056(1071.7)	1052(1060)	1095(1145.8)
g56	1052	**1014(1023)**	1047(1065.5)	1056(1071.7)	1052(1059.6)	1115(1175.3)
g57	100	**100**	**100**	**100**	**100**	**100**(109.6)
g58	1180	**1146(1156.6)**	1190(1219.2)	1195(1213.7)	1160(1173.9)	1239(1314.5)
g59	1180	**1146(1156.6)**	1190(1219.2)	1195(1213.7)	1160(1173.4)	1249(1292.1)
g60	1455	**1402(1417.2)**	1465(1480)	1462(1474.8)	1475(1486.4)	1626(1666.4)
g61	1455	**1402(1417.2)**	1465(1480)	1462(1474.8)	1475(1487)	1602(1663.2)
g62	140	**140**	**140**	**140**	**140**	142(274.1)
g63	1659	**1613(1624.4)**	1680(1700.1)	1659(1696.6)	1688(1702)	1808(1859.7)
g64	1659	**1613(1624.4)**	1680(1700.1)	1659(1696.6)	1688(1697.1)	1826(1867)
g65	160	**160**	**160**	**160**	**160**	213(347.8)
g66	180	200	**180**(182)	**180**	200	257(431.8)
g67	199	200	**199**(199.7)	**199(199.5)**	200	360(522.8)
g70	426	**357(364.1)**	380(392.5)	366(379.1)	457(467.9)	515(530)
g72	199	200	**199**(199.7)	**199(199.6)**	200	413(526.7)
g77	200	**200**	**200**	**200**	**200**	885(1072)
g81	200	**200**	**200**(346.3)	**200**(287.6)	**200**	1950(2110.1)

Table 2: Results for all algorithms on Large benchmark

Instance	Pre_best	CELS max(avg)	BVNSBucket max(avg)	BVNSBucket2 max(avg)	CLHUS max(avg)	BVNS max(avg)
494_bus	6	**6**	**6**	**6**	**6**	**6**
arc130	8	**8**	**8**	**8**	**8**	**8**
ash292	9	**9**	**9**	**9**	**9**	**9**(9.2)
ash85	7	**7**	**7**	**7**	**7**	**7**
bcspwr01	3	**3**	**3**	**3**	**3**	**3**
bcspwr02	2	**2**	**2**	**2**	**2**	**2**
bcspwr03	4	**4**	**4**	**4**	**4**	**4**
bcspwr04	7	**7**	**7**(7.1)	**7**	**7**	**7**(7.2)
bcspwr05	7	**7**	**7**	**7**	**7**	**7**
bcsstk01	12	**12**	**12**	**12**	**12**	**12**
bcsstk04	24	**24**	**24**	**24**	**24**	**24**
bcsstk05	15	**15**	**15**(15.6)	**15**(15.4)	**15**	16
bcsstk06	36	**36**	**36**	**36**	**36**	**36**(36.2)
bcsstk07	36	**36**	**36**	**36**	**36**	**36**(37)
bcsstk20	7	10(12.5)	**7**(7.4)	**7**	10(13.6)	**7**
bcsstk22	4	**4**	**4**	**4**	**4**	**4**
bcsstm07	36	**36**	**36**(37.8)	**36**	**36**	**36**(39.2)
can___24	4	**4**	**4**	**4**	**4**	**4**
can___61	5	**5**	**5**	**5**	**5**	**5**
can___62	3	**3**	**3**	**3**	**3**	**3**
can___73	8	**8**	**8**	**8**	**8**	**8**
can__144	6	**6**	**6**	**6**	**6**	**6**
can__161	16	**16**	**16**	**16**	**16**	**16**
can__292	18	**18**	**18**	**18**	**18**	**18**(18.3)
can__445	38	**38**(38.8)	**38**(39.6)	40	**38**	**38**(39.4)
curtis54	4	**4**	**4**	**4**	**4**	**4**
dwt__162	7	**7**	**7**	**7**	**7**	**7**
dwt__193	23	**23**	**23**	**23**	**23**	**23**
dwt__209	15	**15**	**15**	**15**	**15**	**15**
dwt__221	7	**8**(8.3)	**8**	**8**	**8**	**8**
dwt__234	4	**4**	**4**	**4**	8	**4**
dwt__245	8	**8**	**8**(8.6)	**8**(8.5)	**8**(8.4)	**8**(8.4)
dwt__310	8	**8**	**8**	**8**	**8**	**8**
dwt__361	14	**14**	**14**	**14**	**14**	**14**
dwt__419	16	**16**	**16**	**16**	**16**	**16**
fs_183_1	15	**15**	**15**	**15**	**15**	**15**
fs_183_3	15	**15**	**15**	**15**	**15**	**15**
fs_183_4	15	**15**	**15**	**15**	**15**	**15**
fs_183_6	15	**15**	**15**	**15**	**15**	**15**
gent113	13	**13**	**13**	**13**	**13**	**13**
gre__115	18	**18**	**18**	**18**	**18**	**18**
gre__185	20	**20**	**20**	**20**	**20**	**20**
gre__343	28	**28**	**28**	**28**	**28**	**28**
gre_216a	21	**21**	**21**	**21**	**21**	**21**
gre_216b	21	**21**	**21**	**21**	**21**	**21**
grid_5.Changed	5	**5**	**5**	**5**	**5**	**5**
grid_6.Changed	6	**6**	**6**	**6**	**6**	**6**
grid_7.Changed	7	**7**	**7**	**7**	**7**	**7**
Grid3x3.Changed	3	**3**	**3**	**3**	**3**	**3**
hor__131	33	**33**(33.4)	**33**	**33**	**33**(33.4)	**33**(33.1)
hypercube_4_16	6	**6**	**6**	**6**	**6**	**6**
hypercube_5_32	10	**10**	**10**	**10**	**10**	**10**
hypercube_6_64	20	**20**	**20**	**20**	24	**20**
hypercube_7_128	35	**35**	**35**	**35**	40(40.3)	**35**
hypercube_8_256	70	**70**	**70**	**70**	78(85)	**70**
ibm32	9	**9**	**9**	**9**	**9**	**9**
impcol_a	20	**20**	**20**	**20**	**20**	**20**(20.4)
impcol_b	15	**15**	**15**	**15**	**15**	**15**
impcol_c	21	**21**	**21**	**21**	**21**	**21**
impcol_d	17	**17**(17.9)	**17**(17.7)	**17**(17.7)	**17**	**17**(17.5)
impcol_e	30	**30**	**30**	**30**	**30**	**30**
lns__131	11	**11**	**11**	**11**	**11**	**11**
lund_a	20	**20**	**20**	**20**	**20**	**20**
lund_b	20	**20**	**20**	**20**	**20**	**20**
mbeacxc	187	**187**(187.5)	231(235.6)	232(235)	**187**	206(210)
mbeaflw	187	**187**(187.5)	231(235.6)	232(235)	**187**	207(207.5)
mbeause	178	**178**	206(221.8)	215(228.1)	**178**	180(198.5)
mcca	18	**18**	**18**	**18**	**18**	**18**
nnc261	11	**11**	**11**	**11**	**11**	**11**
nos1	3	**3**	**3**	**3**	**3**	**3**
nos4	7	**7**	**7**	**7**	**7**	**7**
p100_24_34	3	**3**	**3**	**3**	**3**	**3**
p17_16_24	3	**3**	**3**	**3**	4	**3**
p18_16_21	2	**2**	**2**	**2**	3	**2**
p19_16_19	2	**2**	**2**	**2**	**2**(2.2)	**2**
p20_16_18	2	**2**	**2**	**2**	**2**	**2**
p21_17_20	2	**2**	**2**	**2**	**2**	**2**
p22_17_19	2	**2**	**2**	**2**	**2**	**2**
p23_17_23	2	**2**	**2**	**2**	**2**	**2**
p24_17_29	3	**3**	**3**	**3**	**3**(3.6)	**3**
p25_17_20	2	**2**	**2**	**2**	**2**	**2**
p26_17_19	2	**2**	**2**	**2**	3	**2**
p27_17_19	2	**2**	**2**	**2**	**2**	**2**
p28_17_18	2	**2**	**2**	**2**	**2**	**2**
p29_17_18	1	**1**	**1**	**1**	2	**1**
p30_17_19	2	**2**	**2**	**2**	**2**	**2**
p31_18_21	2	**2**	**2**	**2**	3	**2**
p32_18_20	2	**2**	**2**	**2**	**2**	**2**
p33_18_21	3	**3**	**3**	**3**	**3**	**3**

Table 3: Results for all algorithms on classic_medium benchmark I

Instance	Pre_best	CELS max(avg)	BVNSBucket max(avg)	BVNSBucket2 max(avg)	CLHUS max(avg)	BVNS max(avg)
p34_18_21	2	**2**	**2**	**2**	**2**	**2**
p35_18_19	2	**2**	**2**	**2**	**2**	**2**
p36_18_20	2	**2**	**2**	**2**	**2**(2.1)	**2**
p37_18_20	2	**2**	**2**	**2**	**2**	**2**
p38_18_19	2	**2**	**2**	**2**	**2**	**2**
p39_18_19	2	**2**	**2**	**2**	**2**	**2**
p40_18_32	4	**4**	**4**	**4**	**4**	**4**
p41_19_20	1	**1**	**1**	**1**	2	**1**
p42_19_24	3	**3**	**3**	**3**	4	**3**
p43_19_22	2	**2**	**2**	**2**	**2**	**2**
p44_19_25	3	**3**	**3**	**3**	**3**	**3**
p45_19_25	2	**2**	**2**	**2**	3	**2**
p46_19_20	2	**2**	**2**	**2**	**2**	**2**
p47_19_21	2	**2**	**2**	**2**	**2**	**2**
p48_19_21	2	**2**	**2**	**2**	**2**	**2**
p49_19_22	2	**2**	**2**	**2**	**2**	**2**
p50_19_25	2	**2**	**2**	**2**	**2**(2.9)	**2**
p51_20_28	4	**4**	**4**	**4**	**4**	**4**
p52_20_27	2	**2**	**2**	**2**	**2**	**2**
p53_20_22	2	**2**	**2**	**2**	**2**	**2**
p54_20_28	3	**3**	**3**	**3**	**3**(3.3)	**3**
p55_20_24	2	**2**	**2**	**2**	**2**	**2**
p56_20_23	3	**3**	**3**	**3**	**3**	**3**
p57_20_24	2	**2**	**2**	**2**	**2**	**2**
p58_20_21	2	**2**	**2**	**2**	**2**	**2**
p59_20_23	2	**2**	**2**	**2**	**2**(2.2)	**2**
p60_20_22	2	**2**	**2**	**2**	3	**2**
p61_21_22	2	**2**	**2**	**2**	**2**	**2**
p62_21_30	3	**3**	**3**	**3**	**3**(3.3)	**3**
p63_21_42	5	**5**	**5**	**5**	**5**	**5**
p64_21_22	2	**2**	**2**	**2**	**2**	**2**
p65_21_24	2	**2**	**2**	**2**	**2**	**2**
p66_21_28	3	**3**	**3**	**3**	**3**	**3**
p67_21_22	2	**2**	**2**	**2**	**2**	**2**
p68_21_27	3	**3**	**3**	**3**	**3**	**3**
p69_21_23	2	**2**	**2**	**2**	**2**	**2**
p70_21_25	3	**3**	**3**	**3**	**3**	**3**
p71_22_29	3	**3**	**3**	**3**	**3**	**3**
p72_22_49	5	**5**	**5**	**5**	**5**	**5**
p73_22_29	2	**2**	**2**	**2**	**2**	**2**
p74_22_30	3	**3**	**3**	**3**	**3**	**3**
p75_22_25	2	**2**	**2**	**2**	**2**	**2**
p76_22_30	2	**2**	**2**	**2**	**2**	**2**
p77_22_37	4	**4**	**4**	**4**	**4**	**4**
p78_22_31	3	**3**	**3**	**3**	**3**	**3**
p79_22_29	3	**3**	**3**	**3**	**3**	**3**
p80_22_30	3	**3**	**3**	**3**	**3**	**3**
p81_23_46	6	**6**	**6**	**6**	**6**	**6**
p82_23_24	2	**2**	**2**	**2**	**2**	**2**
p83_23_24	1	**1**	**1**	**1**	**1**	**1**
p84_23_26	2	**2**	**2**	**2**	**2**	**2**
p85_23_26	1	**1**	**1**	**1**	**1**	**1**
p86_23_24	2	**2**	**2**	**2**	**2**	**2**
p87_23_30	3	**3**	**3**	**3**	**3**	**3**
p88_23_26	2	**2**	**2**	**2**	**2**	**2**
p89_23_27	3	**3**	**3**	**3**	**3**	**3**
p90_23_35	3	**3**	**3**	**3**	**3**	**3**
p91_24_33	3	**3**	**3**	**3**	**3**	**3**
p92_24_26	2	**2**	**2**	**2**	**2**	**2**
p93_24_27	2	**2**	**2**	**2**	**2**	**2**
p94_24_31	3	**3**	**3**	**3**	**3**	**3**
p95_24_27	2	**2**	**2**	**2**	**2**	**2**
p96_24_27	2	**2**	**2**	**2**	3	**2**
p97_24_26	2	**2**	**2**	**2**	**2**	**2**
p98_24_29	2	**2**	**2**	**2**	**2**	**2**
p99_24_27	2	**2**	**2**	**2**	**2**	**2**
plat362	27	**27**	**27**	**27**	**27**	**27**
plskz362	10	**10**	**10**	**10**	**10**	**10**
pores_1	7	**7**	**7**	**7**	**7**	**7**
pores_3	12	**12**	**12**	**12**	**12**	**12**
saylr1	14	**14**	**14**	**14**	**14**(15.9)	**14**
steam1	39	**39**	**39**	**39**	**39**	**39**
steam3	4	**4**	**4**	**4**	**4**	**4**
str____0	81	**81**	83(84.8)	83(84.9)	**81**(81.4)	82(85)
str__200	95	**94(94.6)**	95(95.8)	95(95.7)	95	95(96.3)
str__600	102	**101(101.2)**	103(104.3)	102(103.4)	102(102.8)	102(103.9)
TREE_22_3_rot1.Changed	2	**2**	**2**	**2**	**2**	**2**
TREE_22_3_rot2.Changed	2	**2**	**2**	**2**	**2**	**2**
TREE_22_3_rot3.Changed	2	**2**	**2**	**2**	**2**	**2**
TREE_22_3_rot4.Changed	2	**2**	**2**	**2**	**2**	**2**
TREE_22_3_rot5.Changed	2	**2**	**2**	**2**	**2**	**2**
west0132	18	**18**	**18**	**18**	**18**	**18**
west0156	26	**26**	**26**	**26**	**26**	**26**
west0167	19	**19**	**19**	**19**	**19**	**19**
west0381	110	**110**	111(112.1)	111(111.8)	**110**(110.2)	111(112.3)
west0479	75	**75**(75.1)	76(76.2)	**75**(75.8)	**75**	76(76.4)
west0497	44	**44**(47.6)	**44**	**44**	**44**(44.7)	**44**(47.5)
will199	52	**52**	**52**	**52**	**52**	**52**
will57	3	**3**	**3**	**3**	**3**	**3**

Table 4: Results for all algorithms on classic_medium benchmark II

Instance	Pre_best	CELS max(avg)	BVNSBucket max(avg)	BVNSBucket2 max(avg)	CLHUS max(avg)	BVNS max(avg)
685_bus	8	**8**	**8**(8.8)	**8**(8.5)	**8**(8.6)	**8**(8.5)
bcsstk08	60	**60**	**60**(60.9)	**60**(61)	**60**(61.5)	**60**(60.9)
bcsstk09	61	**61**	**61**	**61**	**61**	**61**
bcsstk11	36	**36**	**36**(85.3)	**36**(75.9)	**36**	92(160.2)
bcsstk12	36	**36**	**36**(85.3)	**36**(75.9)	**36**	91(173.3)
bcsstk19	4	**4**	**4**	**4**	**4**	**4**(8.2)
bcsstk27	41	**41**	45(92.1)	**41**(92.7)	**41**(43)	425(524.2)
bcsstm27	41	**41**	45(92.1)	**41**(92.7)	**41**(43)	483(532.2)
blckhole	61	**61**	62(62.9)	62(62.9)	**61**(62.3)	62(63)
bp_____0	153	**150(150.9)**	155(156.5)	152(156.5)	153(153.6)	155(157.2)
bp___200	175	**174(175.3)**	181(182.3)	181(183.5)	175(176.1)	182(186.2)
bp___400	184	**183**(185.5)	188(193.6)	186(192.3)	185(**185.3**)	186(193.9)
bp___600	195	193(**194**)	195(201.1)	**191**(199.2)	196(197)	197(202)
bp___800	199	**198(198.3)**	202(205.6)	200(204.9)	199(199.6)	203(206.7)
bp__1000	203	**202(202.1)**	204(209.7)	204(210.2)	203(203.7)	208(210.4)
bp__1200	207	**207**	211(215.4)	209(213.5)	208(208.9)	212(216.2)
bp__1400	211	**210**	211(217.6)	212(216.1)	211(211.9)	216(219.8)
bp__1600	211	**209(209.7)**	213(217.1)	213(217.2)	211(211.8)	215(218.1)
can__715	35	**35**	**35**(35.6)	**35**(37.4)	**35**(35.6)	**35**(38)
can__838	34	**34**(36.3)	**34(34.3)**	**34**(34.4)	**34**(34.8)	**34**(35)
can_1054	28	**28**(35.1)	**28**(30)	**28(29.2)**	30(32.2)	**28**(38.6)
can_1072	30	**30**(30.8)	**30(30.6)**	**30**(30.6)	**30**(31.1)	**30**(33.7)
dwt__503	26	**26**	**26**(32.2)	**26**(31)	**26**	**26**(31.2)
dwt__592	22	**22**(22.2)	**22**(22.2)	**22(22.4)**	**22**	**22**(22.4)
dwt__878	18	**18**	**18**	**18**	**18**	**18**
dwt__918	22	**22**(22.8)	**22**	**22**	**22**	**22**
dwt__992	34	**34**	**34**	**34**	**34**	**34**(34.2)
dwt_1005	33	**33**	**33**	**33**	**33**	**33**
dwt_2680	29	**29**(49)	**29**(37.7)	**29(34.2)**	**29**(39.4)	**29**(82.6)
fs_541_1	19	**19**	**19**	**19**	**19**	**19**
fs_541_2	19	**19**	**19**	**19**	**19**	**19**
fs_541_3	19	**19**	**19**	**19**	**19**	**19**
fs_541_4	19	**19**	**19**	**19**	**19**	**19**
fs_680_1	6	**6**	**6**	**6**	**6**	**6**
fs_680_2	6	**6**	**6**	**6**	**6**	**6**
fs_680_3	6	**6**	**6**	**6**	**6**	**6**
fs_760_1	22	**22**	**22**	**22**	**22**	**22**
fs_760_2	22	**22**	**22**	**22**	**22**	**22**
fs_760_3	22	**22**	**22**	**22**	**22**	**22**
gr_30_30	30	**30**	**30**	**30**	**30**	**30**
gre__512	36	**36**	**36**	**36**	**36**	**36**
gre_1107	90	**90**(93.4)	**90**(91)	**90**(91.2)	**90(90.2)**	**90**(92)
hypercube_10_1024.txt	252	**252**	**252**	**252**	312(354.7)	**252**
hypercube_9_512.txt	126	**126**	**126**	**126**	157(169.4)	**126**
jagmesh1	26	**26**	**26**	**26**	**26**	**26**
jagmesh2	31	**31**	**31**	**31**	**31**	**31**
jagmesh3	33	**33**	**33**	**33**	**33**	**33**
jagmesh7	14	**14**(14.4)	**14**(14.3)	**14(14.1)**	**14**(16.3)	**14**(14.5)
jpwh_991	63	**63**(67.6)	**63(64)**	**63**(64.2)	**63**(64.8)	**63(63.8)**
lns__511	31	**31**(31.4)	**31**(31.3)	**31**	**31**(31.8)	**31**(31.5)
lshp1009	31	**31**	**31**	**31**	**31**	**31**
mcfe	89	**89(89.1)**	104(128.8)	106(126.7)	**89**(89.2)	106(143.9)
nnc666	18	**18**	**18**	**18**	**18**	**18**(19)
nos2	3	**3**(4.2)	**3(3.4)**	**3**(3.8)	**3**(3.6)	**3**(5)
nos3	40	**40**	**40**	**40**	**40**	**40**(44)
nos6	15	**15**	**15**	**15**	**15**	**15**
nos7	65	**65**	**65**	**65**	**65**(66.4)	**65**
orsirr_2	50	**50(50.7)**	**50**(55.4)	**50**(54.8)	51	**50**(53.2)
saylr3	30	**30**(30.7)	**30(30.3)**	**30(30.3)**	34(58.3)	**30**(30.6)
sherman1	30	**30**(30.7)	**30(30.3)**	**30(30.3)**	34(58.3)	**30**(30.4)
sherman4	22	**22**	**22**(23.4)	**22**(22.8)	31	**22**(22.8)
shl____0	82	**80(80.6)**	82(82.3)	82	82	82
shl__200	90	**88(88.5)**	89(90.5)	89(90.8)	90(91)	90(90.8)
steam2	60	**60**	**60**	**60**	**60**	**60**
west0655	108	**108**	**108**(109.5)	**108**(108.8)	**108**(109.2)	**108**(110.2)

Table 5: Results for all algorithms on classic_large benchmark