Solving the Maximum-Weight Connected Subgraph Problem to Optimality

Mohammed El-Kebir[1,2,3] and Gunnar W. Klau[2,3]

[1] Department of Computer Science and Center for Computational Molecular Biology, Brown University, Providence, RI, 02906, USA
[2] Life Sciences group, Centrum Wiskunde & Informatica (CWI), Science Park 123, 1098 XG Amsterdam, the Netherlands
[3] Centre for Integrative Bioinformatics VU (IBIVU), VU University Amsterdam, De Boelelaan 1081A, 1081 HV Amsterdam, the Netherlands
`melkebir@cs.brown.edu, gunnar.klau@cwi.nl`

Abstract. Given an undirected node-weighted graph, the Maximum-Weight Connected Subgraph problem (MWCS) is to identify a subset of nodes of maximal sum of weights that induce a connected subgraph. MWCS is closely related to the well-studied Prize-Collecting Steiner Tree problem and has many applications in different areas, including computational biology, network design and computer vision. The problem is NP-hard and even hard to approximate within a constant factor. In this work we describe an algorithmic scheme for solving MWCS to provable optimality, which is based on preprocessing rules, new results on decomposing an instance into its biconnected and triconnected components and a branch-and-cut approach combined with a primal heuristic. We demonstrate the performance of our method on the benchmark instances of the 11th DIMACS implementation challenge consisting of MWCS as well as transformed PCST instances.

Keywords: maximum-weight connected subgraph, algorithm engineering, divide-and-conquer, SPQR tree, prize-collecting Steiner tree, branch-and-cut

1 Introduction

We consider the Maximum-Weight Connected Subgraph problem (MWCS). Given an undirected node-weighted graph, the task is to find a subset of nodes of maximal sum of weights that induce a connected subgraph. A formal definition of the unrooted and rooted variant is as follows.

Definition 1 (MWCS). *Given an undirected graph $G = (V, E)$ with node weights $w : V \to \mathbb{R}$, find a subset $V^* \subseteq V$ such that the induced graph $G[V^*] := \left(V^*, E \cap \binom{V^*}{2}\right)$ is connected and the weight $w(G[V^*]) := \sum_{v \in V^*} w(v)$ is maximal.*

Definition 2 (R-MWCS). *Given an undirected graph $G = (V, E)$, a node set $R \subseteq V$ and node weights $w : V \to \mathbb{R}$, find a subset $V^* \subseteq V$ such that $R \subseteq V^*$, the induced graph $G[V^*] := \left(V^*, E \cap \binom{V^*}{2}\right)$ is connected and the weight $w(G[V^*]) := \sum_{v \in V^*} w(v)$ is maximal.*

Johnson mentioned MWCS in his NP-completeness column [14]. The problem and its cardinality-constrained and budget variants have numerous important applications in different areas, including designing fiber-optic networks [17], oil-drilling [12], systems biology [2,9,19], wildlife corridor design [8], computer vision [5] and forest planning [4].

The maximum-weight connected subgraph problem is closely related to the well-studied Prize-Collecting Steiner Tree problem (PCST) [15,18], which is defined as follows.

Definition 3 (PCST). *Given an undirected graph $G = (V, E)$ with node profits $p : V \to \mathbb{R}_{\geq 0}$ and edge costs $c : E \to \mathbb{R}_{\geq 0}$, find a connected subgraph $T = (V^*, E^*)$ of G such that $p(T) := \sum_{v \in V^*} p(v) - \sum_{e \in E^*} c(e)$ is maximal.*

In [9] we described a reduction from MWCS to PCST and showed that a prize-collecting Steiner tree T in the transformed instance is a connected subgraph in the original instance with weight $p(T) - w'$, where w' is the minimum weight of a node. We also gave a simple approximation-preserving reduction from PCST to MWCS: Given an instance $(G = (V, E), p, c)$ of PCST, the corresponding instance (G', w) of MWCS is obtained by splitting each edge (v, w) in E into two edges (v, u) and (u, w), and setting the weight $w(u)$ of the introduced split vertex u to $-c(e)$.

Theorem 1. *A maximum-weight connected subgraph T' in the transformed instance corresponds to an optimal prize-collecting Steiner tree T in the original instance, and $w(T') = p(T)$.*

Proof. We first observe that if a split vertex u is part of T', then also its neighbors v and w must be in T', otherwise $T' \setminus \{u\}$ would be a better solution. We then can simply map each split vertex back to its original edge. The solution clearly has profit $p(T) = w(T')$ and is optimal, because a more profitable subgraph with respect to p would also correspond to a higher-scoring subgraph with respect to w, contradicting the optimality of T'. $\square$

These reductions directly imply and simplify a number of results for MWCS. For example, it follows from [11] and Theorem 1 that MWCS is NP-hard and even hard to approximate within a constant factor. In addition, the results in [3] provide a polynomial-time exact algorithm for MWCS for graphs of bounded treewidth.

In [9] we used the close relation to PCST to develop an exact algorithm for MWCS by running the branch-and-cut approach of Ljubic et al. [1] on the transformed instance. Backes et al. [2] presented a direct integer linear programming formulation for a variant of MWCS based only on node variables. Álvarez-Miranda et al. [1] recently introduced a stronger formulation based on the concept of node-separators.

Here, we introduce an algorithm engineering approach that combines existing and new results to solve MWCS instances efficiently in practice to provable optimality. We describe new and adapted preprocessing rules in Section 2. Section 3 is dedicated to an overall divide-and-conquer scheme, which is based on novel results on decomposing an instance into its biconnected and triconnected components. In Section 4 we describe a branch-and-cut approach using a new primal heuristic based on an exact dynamic programming algorithm for trees. We demonstrate in Section 5 the performance of our approach and the benefits of preprocessing and the divide-and-conquer scheme.

2 Preprocessing

We describe reduction rules that simplify an instance of MWCS without losing optimality. We define three classes of increasingly complex reduction rules and apply them exhaustively in successive phases of a preprocessing scheme, see Figure 1.

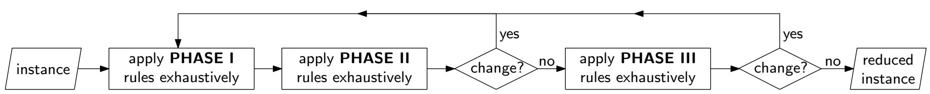

Fig. 1. Preprocessing scheme. An MWCS instance passes through three phases of increasingly complex rules that are run exhaustively until no rules apply anymore. The result is a reduced instance.

The rules make use of three operations on node sets: Merge, Isolate and Remove, see Figure 2. Given a node set V', Merge (V') combines the nodes in V' into a supernode of weight $\sum_{v \in V'} w(v)$, which is connected to all neighbors of nodes in V' outside V'. Operation Isolate (V') adds a copy of V' without edges and merges it. Operation Remove (V') removes all nodes in V' from the graph. We keep a mapping from the merged nodes to sets of original nodes to map solutions of the reduced instance to solutions of the original instance. These operations will also be used in our divide-and-conquer scheme, which we will present in Section 3.

- **Phase I rules.** The first phase consists of three simple rules.
  1. *Remove isolated negative node rule.* Let v be an isolated vertex with $w(v) < 0$. We can safely remove v by calling Remove ($\{v\}$), because it will never be part of any optimal solution. Identifying all nodes that satisfy the condition takes $O(|V|)$ time.

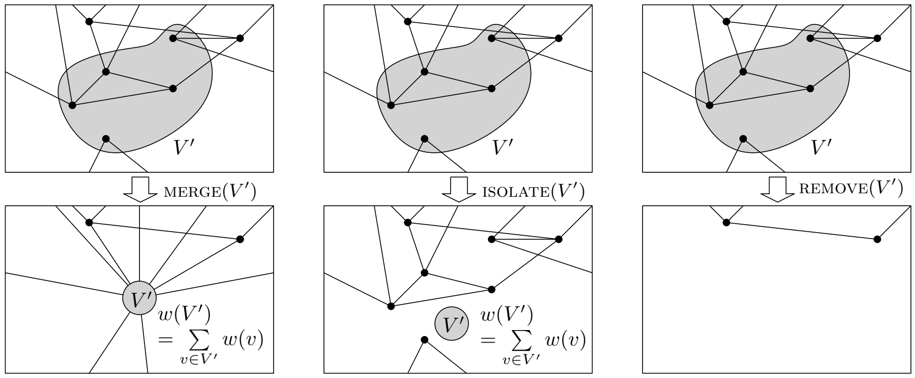

Fig. 2. Operations Merge, Isolate, Remove.

2. *Merge adjacent positive nodes rule.* Let (u, v) be an edge with $w(u) > 0$ and $w(v) > 0$. If one vertex will be part of the solution the other one will be as well, so we perform Merge $(\{u, v\})$. Finding all adjacent positively-weighted nodes takes $O(|E|)$ time.
3. *Merge negative chain rule.* Let P be a chain of negative degree 2 vertices. It is safe to perform Merge (P). Either none of the vertices in P will be part of an optimal solution or all of them. In the latter case P is used as a bridge between positive parts. Identifying all negatively-weighted chains takes $O(|E|)$ time.

– **Phase II rules.** The second phase consists of one rule.

1. *Mirrored hubs rule.* Let $u, v \in V$ be two distinct negatively-weighted nodes, i.e, $w(u) < 0$ and $w(v) < 0$. Without loss of generality assume that $w(u) \leq w(v)$. If u and v are adjacent to the same nodes then we can Remove $(\{u\})$. The reason is that v will always be preferred over u in an optimal solution, because it is adjacent to exactly the same nodes as u and costs less. Finding all pairs of negatively-weighted mirrored nodes takes $O(\Delta \cdot |V|^2)$ time where Δ is the maximum degree of the graph.

– **Phase III rule.** The last phase consists of the most expensive rule.

1. *Least-cost rule.* This rule is adapted from the least-cost test, which was described by Duin and Volgenant [10] for the node-weighted Steiner tree problem. Let (u, v) and (v, w) be two edges in the graph, and let v have degree 2 and $w(v) < 0$. We construct a directed graph whose node set is V and whose arc set A is obtained by introducing for every edge (a, b) in G two oppositely directed arcs ab and ba. We can Remove $(\{v\})$, if the shortest path from u to w with respect to lengths $d(ab) := \max\{-w(b), 0\}$ for all $ab \in A$ is shorter than $-w(v)$. The reason is that if u and w were to be in an optimal solution there is a better way than using v. This rule takes $O(|V'| \cdot (|E| + |V| \log |V|))$ time where V' is the set of all negative-weighted nodes having degree 2.

Algorithm 1: SolveMWCS($G = (V, E), w$)

foreach connected component C of G **do**
  Preprocess (C)
  let T_{B} be the block-cut vertex tree of C
  while T_B has block B of degree 0 or 1 **do**
    ProcessBicomponent (B)
    update T_B
  V_C = SolveUnrooted (C)
  Merge (V_C); Remove ($C \setminus V_c$)
$V^* \leftarrow$ SolveUnrooted (G)
return V^*

3 Divide-and-Conquer Scheme

We propose a three-layer divide-and-conquer scheme for solving MWCS to provable optimality. It is based on decomposing the input graph into its connected, biconnected and triconnected components. Hüffner et al. have already considered data reduction rules based on heuristically found separators of size k for the Balanced Subgraph problem [13]. Here, we present the first data reduction approach that considers all separators of size 1 and 2 in a rigorous manner by processing them using the block-cut and SPQR tree data structures.

In the first layer, we consider the connected components of the input (G, w) one-by-one, see Algorithm 1. In the next layer, we construct a block-cut vertex tree T_{B} for each connected component C. We process the block leaves B of T_{B} iteratively. Processing a block B of degree 1 will result in the removal of $B \setminus \{c\}$, where c is the corresponding cut vertex. In addition, a new degree 0 node may be introduced. Processing a block B of degree 0 will result in the replacement of B by a single isolated node. Therefore, at the end of the loop, the graph $G[C]$ will only consist of isolated nodes. Among these nodes, the node with maximum weight corresponds to the maximum weight connected subgraph of $G[C]$. We retain only this node in the graph, and remove all other nodes in C. After processing all connected components, a similar situation arises in G: each component is an isolated node, and the solution V^* will correspond to the node that has maximum weight.

Next, we describe how to process a block B. The idea here is to account for the situation where the final optimal solution V^* contains parts of B, i.e. $V^* \cap B \neq \emptyset$. For this to happen, either V^* must be a proper subset of B, or a cut node of B must be part of V^*. Since B corresponds to a degree 0 or 1 block in T_{B}, it contains at most one cut node c. Let us consider the case where B does have a cut node c, as the other case is straightforwardly resolved by introducing an isolated node. Two subcases can be distinguished: $c \in V^* \cap B$ and $c \notin V^* \cap B$. We encode both cases using the following gadget. Let V_1 be the unrooted maximum-weight connected subgraph of $G[B]$, and let V_2 be the maximum-weight connected subgraph of $G[B]$ rooted at c. The corresponding gadget Γ_1 is obtained by merging the nodes in V_2, and, if $V_1 \neq V_2$, by additionally introducing an isolated vertex corresponding to V_1—see Figure 3 D and E. Replacing B by the gadget preserves optimality as stated in the following lemma.

Lemma 1. *Let $B \subseteq V$ be a block in $G = (V, E)$ containing exactly one cut node c. Let $G' = G[(V \setminus B) \cup \Gamma_1]$ be the graph where B is replaced by gadget Γ_1. A maximum weight connected subgraph of $G'[U^*]$ has the same weight as a maximum weight connected subgraph $G[V^*]$, i.e., $w(U^*) = w(V^*)$.*

Proof. The gadget Γ_1 consists of two parts V_1 and V_2, which correspond to the unrooted and $\{c\}$-rooted maximum weight connected subgraph of $G[B]$, respectively. By definition V_1 and V_2 induce connected subgraphs in G. Therefore the operations Merge (V_2) and Isolate (V_1)—resulting in the construction of Γ_1—combined with the optimality of V^* ensure that $w(U^*) \leq w(V^*)$.

We now distinguish two subcases: $V^* \cap B = \emptyset$ and $V^* \cap B \neq \emptyset$. Consider the first case. Since the introduction of the gadget only concerns nodes in B, we have that $w(U^*) \geq w(V^*)$. Hence, $w(U^*) = w(V^*)$.

In the other case, $V^* \cap B \neq \emptyset$, we either have that $c \notin V^* \cap B$ or $c \in V^* \cap B$. If $c \notin V^* \cap B$ then $V^* \subseteq B$. By construction of the gadget, we then have $w(V_1) = w(V^* \cap B) = w(V^*)$. Conversely, if $c \in V^* \cap B$ then $w(V_2) = w(V^* \cap B)$. Observe that $w(U^* \setminus \Gamma_1) = w(V^* \setminus B)$. Therefore $w(U^*) = w(V^*)$. □

As an optimization, we preemptively remove a leaf block B if all its nodes $v \in B \setminus \{c\}$ have nonpositive weights $w(v) \leq 0$.

In the third layer, we start by constructing an SPQR-tree T_{SPQR} of B. We then iteratively consider each triconnected component A that does not contain the cut node c and contains at least three nodes. Let $\{u, v\}$ be the cut pair of such a triconnected component A. If A consists of only negatively weighted nodes, its only purpose is to connect u with v. To find the cheapest way of doing this, we construct a directed graph whose node set is A and whose arcs are obtained by introducing for every edge (a, b) in $G[A]$ two oppositely directed arcs. We define the cost of an arc (a, b) to be $-w(b)$. The cheapest way of going from u to v now corresponds to the shortest path from u to v in the directed graph. Triconnected components that contain positively-weighted nodes are processed separately and may be replaced by gadgets of smaller size, which we describe next.

Let us consider the situation where the final solution V^* contains parts of a triconnected component A with cut nodes $\{u, v\}$, i.e., $V^* \cap A \neq \emptyset$. We can distinguish four cases: (i) $u \in V^*$, (ii) $v \in V^*$, (iii) $\{u, v\} \subseteq V^*$, and (iv) $V^* \subseteq A$. In the following we introduce a gadget Γ_2 that encodes all four cases. The first three cases correspond to finding a rooted maximum weighted connected subgraph in $G[A]$ with $\{u\}$, $\{v\}$ and $\{u, v\}$ as the root node sets, respectively. Let V_1, V_2, V_3 be the solutions sets of the three rooted maximum weight connected problems from which the respective root nodes have been removed. The fourth case corresponds to finding an unrooted maximum weight connected subgraph in $G[A]$ whose solution we denote by V_4. To encode the fourth case, we Isolate set V_4. As for the first three cases, we Merge the sets $V_1 \setminus V_2$, $V_2 \setminus V_1$, $V_1 \cap V_2$ and $V_3 \setminus (V_1 \cup V_2)$ resulting in the nodes v_1, v_2, v_3 and v_4, respectively. As some

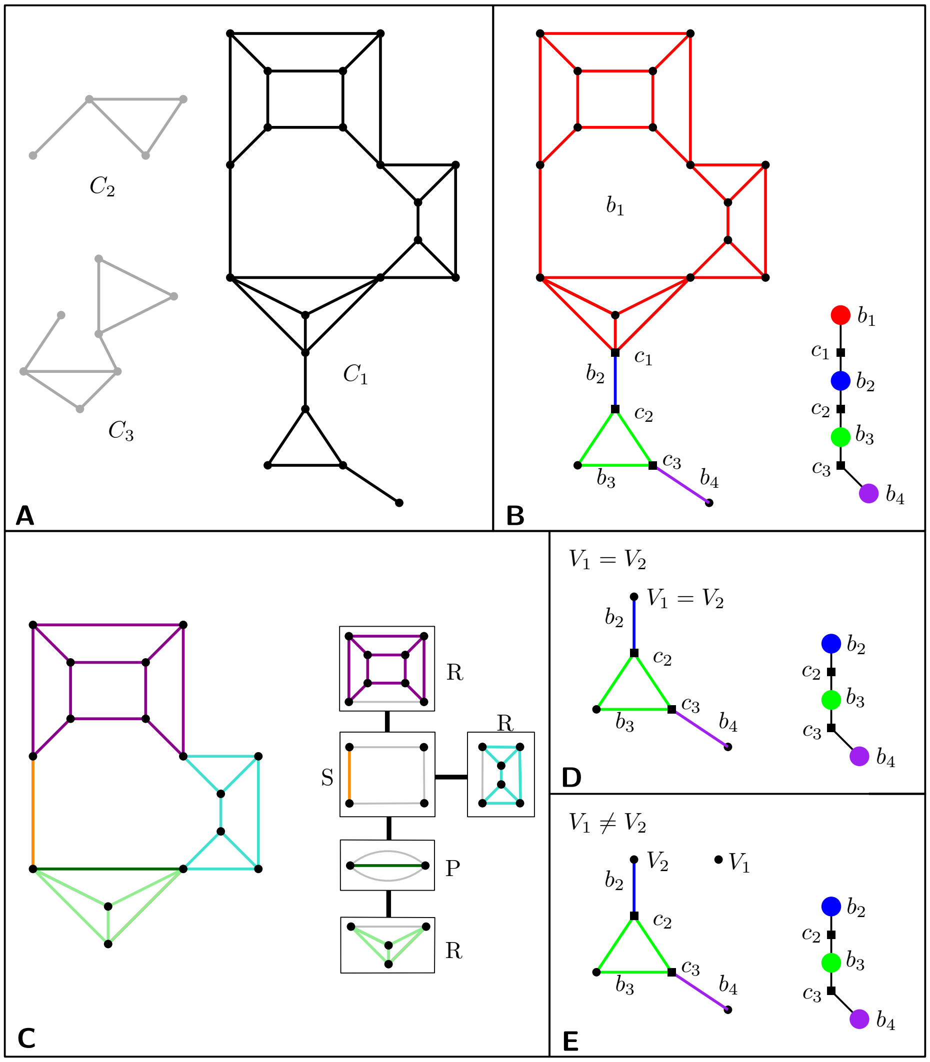

Fig. 3. The three layers of the divide-and-conquer scheme. A: Three connected components of an MWCS instance. **B:** Biconnected components and the block-cut vertex tree of connected component C_1. **C:** Triconnected components and the SPQR tree of biconnected component b_1. **D:** Gadget Γ_1 in the first case. **E:** Gadget Γ_1 in the second case.

of these sets may be empty, we need to take care when connecting the gadget. For instance, if $V_1 \setminus V_2 = \emptyset$ and $V_1 \cap V_2 \neq \emptyset$ then we need to connect u directly with v_3. Also, we ensure that we do not break biconnectivity. For instance, if $V_1 \cap V_2 = \emptyset$ and $V_1 \neq \emptyset$ then we merge v_1 and u as to prevent u from becoming an articulation point. See Figure 4 and the pseudocode below for more details.

Procedure ProcessBicomponent(B)

let c be the corresponding cut node, if applicable
if all v in $B \setminus \{c\}$ have $w(v) \leq 0$ **then** Remove ($B \setminus \{c\}$)
else
    let T_{SPQR} be the SPQR tree of B
    foreach triconnected component A of size > 3 not containing c **do**
        let $\{u, v\}$ be the cut pair of A
        if all v in A have $w(v) \leq 0$ **then**
            compute shortest path P from u to v
            Merge ($P \setminus \{u, v\}$); Remove ($N \setminus P$)
        else
            ProcessTricomponent (A)
        Preprocess (B); update T_{SPQR}
    $V_1 \leftarrow$ SolveUnrooted (B)
    $V_2 \leftarrow$ SolveRooted ($B, \{c\}$)
    if $V_1 = V_2$ **then** Merge (V_2); Remove ($B \setminus V_2$)
    else Isolate (V_1); Merge (V_2); Remove ($B \setminus V_2$)

Procedure ProcessTriComponent(A)

let $\{u, v\}$ be the cut pair
$V_1 \leftarrow$ SolveRooted (A, $\{u\}$) $\setminus\{u\}$
$V_2 \leftarrow$ SolveRooted (A, $\{v\}$) $\setminus\{v\}$
$V_3 \leftarrow$ SolveRooted (A, $\{u, v\}$) $\setminus\{u, v\}$
$V_4 \leftarrow$ SolveUnrooted (A)
Isolate (V_4)
$\Gamma_2 \leftarrow \{u, v\}$
if $V_1 \setminus V_2 \neq \emptyset$ **then** $v_1 \leftarrow$ Merge ($V_1 \setminus V_2$); add edge (u, v_1); add v_1 to Γ_2
if $V_2 \setminus V_1 \neq \emptyset$ **then** $v_2 \leftarrow$ Merge ($V_2 \setminus V_1$); add edge (v, v_2); add v_2 to Γ_2
if $V_1 \cap V_2 = \emptyset$ **then**
    if $V_1 \neq \emptyset$ **then** Merge ($\{u, v_1\}$); remove v_1 from Γ_2
    if $V_2 \neq \emptyset$ **then** Merge ($\{v, v_2\}$); remove v_2 from Γ_2
else
    $v_3 \leftarrow$ Merge ($V_1 \cap V_2$); add v_3 to Γ_2
    if $V_1 \subseteq V_2$ **then** add edge (u, v_3) **else** add edge (v_1, v_3)
    if $V_2 \subseteq V_1$ **then** add edge (v, v_3) **else** add edge (v_2, v_3)
if $V_3 \setminus (V_1 \cup V_2) \neq \emptyset$ **then**
    $v_4 \leftarrow$ Merge ($V_3 \setminus (V_1 \cup V_2)$)
    add v_4 to Γ_2
    add edges $(u, v_4), (v, v_4)$
if $V_1 \cap V_2 = \emptyset$ and $V_3 \setminus (V_1 \cup V_2) = \emptyset$ **then**
    if $V_1 \setminus V_2 \neq \emptyset$ **then** add edge (v_1, v)
    if $V_2 \setminus V_1 \neq \emptyset$ **then** add edge (v_2, u)
Remove ($A \setminus \Gamma_2$)

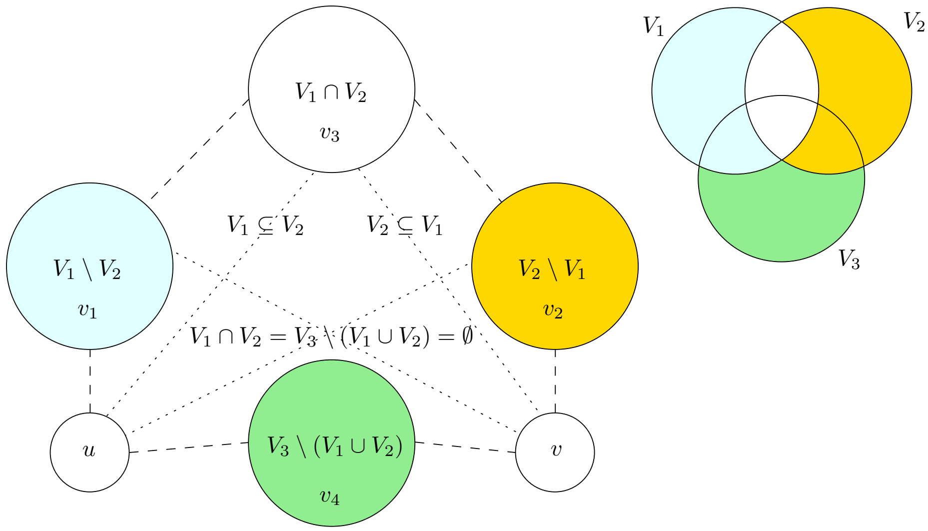

Fig. 4. Triconnected component gadget. The gadget Γ_2 consists of at most four nodes. In case the corresponding set is empty no node is introduced. The dotted edges are only introduced if the condition on the edge is met, e.g., there is an edge from u to v_3 if $V_1 \subsetneq V_2$ and $V_1 \cap V_2 \neq \emptyset$.

Lemma 2. *Let $A \subseteq V$ be a triconnected component in $G = (V, E)$ not containing any cut node of G. Let $G' = G[(V \setminus B) \cup \Gamma_2]$ be the graph where A is replaced by gadget Γ_2. A maximum weight connected subgraph of $G'[U^*]$ has the same weight as a maximum weight connected subgraph $G[V^*]$, i.e., $w(U^*) = w(V^*)$.*

Proof. Let $\{u, v\}$ be the cut pair of A. The gadget Γ_2 encodes four node sets: V_1, V_2 and V_3 representing the rooted maximum weight connected subgraphs of $G[A]$—without their respective root nodes—rooted at $\{u\}$, $\{v\}$ and $\{u, v\}$, respectively; and V_4 representing the unrooted maximum weight connected subgraph of $G[A]$. Let $v_1 := \textsc{Merge}(V_1 \setminus V_2)$, $v_2 := \textsc{Merge}(V_2 \setminus V_1)$, $v_3 := V_1 \cap V_2$ and $v_4 := V_3 \setminus (V_1 \cup V_2)$—see Figure 4.

We start by proving $w(U^*) \leq w(V^*)$. Since A is a triconnected component, we have that $V_1 \setminus V_2$, $V_2 \setminus V_1$, $V_1 \cap V_2$ and $V_3 \setminus (V_1 \cup V_2)$ are connected in G. In addition, as these node sets are obtained by $\textsc{Merge}$ operations only and they are pairwise disjoint, we have that $w(U^*) \leq w(V^*)$.

We distinguish two cases: $V^* \cap A = \emptyset$ and $V^* \cap A \neq \emptyset$. The first case holds, because introduction of the gadget Γ_2 only concerns nodes in A. Therefore, $w(U^*) \geq w(V^*)$, which implies $w(U^*) = w(V^*)$. The second case, $V^* \cap A \neq \emptyset$, has the following four subcases:

1. $u \notin V^*$ and $v \notin V^*$;
   This implies that $V^* \subseteq B$. We then have $w(V_4) = w(V^* \cap A) = w(V^*)$.

2. $u \in V^*$ and $v \notin V^*$;
   By optimality of V^*, we have that $w(V_1 \cup \{u\}) = w(u) + w(v_1) + w(v_3) = w(V^* \cap A)$. Since $w(U^*) \leq w(V^*)$, it follows that $w(U^*) = w(V^*)$.
3. $u \notin V^*$ and $v \in V^*$;
   Symmetric to previous subcase.
4. $u \in V^*$ and $v \in V^*$;
   There are two cases: $V_1 \cap V_2 = \emptyset$ or $V_1 \cap V_2 \neq \emptyset$. In the first case, we have that $w(V_3 \cup \{u, v\}) = w(u) + w(v) + w(v_1) + w(v_2) + w(v_4) = w(V^* \cap A)$. In the second case, we have that $w(V_3 \cup \{u, v\}) = w(u) + w(v) + w(v_1) + w(v_2) + w(v_3) = w(V^* \cap A)$. Since $w(U^*) \leq w(V^*)$, it follows in both cases that $w(U^*) = w(V^*)$. $\square$

Lemmas 1 and 2 imply the correctness of our divide-and-conquer scheme.

Theorem 2. *Given an instance of MWCS Algorithm 1 returns an optimal solution.*

4 Branch-and-Cut Algorithm

To solve the nontrivial instances within our divide-and-conquer scheme, we use a branch-and-cut approach. We obtain strong upper bounds from solving the linear programming (LP) relaxation of an integer linear programming formulation and lower bounds from an integrated primal heuristics that is guided by the optimal solution of the LP relaxation.

4.1 Integer linear programming formulation

We use a formulation that only used node variables for both the unrooted and the rooted MWCS problem. The formulations are equivalent to the generalized node-separator formulation described in [1].

Unrooted. Variables $\mathbf{x} \in \{0,1\}^V$ encode the presence of a node in the solution. To encode connectivity in the unrooted case, we use auxiliary variables $\mathbf{y} \in \{0,1\}^V$ that encode the presence of the root node. The ILP is as follows.

$$\max \sum_{v \in V} w_v x_v \tag{1}$$

$$\sum_{v \in V} y_v = 1 \tag{2}$$

$$y_v \leq x_v \qquad \forall v \in V \tag{3}$$

$$x_v \leq \sum_{u \in \delta(S)} x_u + \sum_{u \in S} y_u \qquad \forall v \in V, \{v\} \subseteq S \subseteq V \tag{4}$$

$$x_v \in \{0,1\} \qquad \forall v \in V \tag{5}$$

$$y_v \in \{0,1\} \qquad \forall v \in V \tag{6}$$

Constraint (2) states that there is exactly one root node. A node can only be the root node if it is present in the solution, which is captured by constraints (3). Constraints (4) state that a node v can only be present in the solution if for all sets S containing v, either the root node is in S, or a node in the set $\delta(S) = \{v \in V \setminus S \mid \exists u \in S : (u,v) \in E\}$ is in the solution. In the next subsection we describe how we separate these constraints.

To strengthen the formulation, we use the following additional cuts.

$$y_v = 0 \qquad \forall v \in V, w(v) < 0 \tag{7}$$

$$\sum_{v>u} y_v \leq 1 - x_u \qquad \forall u \in V, w(u) > 0 \tag{8}$$

$$x_v \leq x_u \qquad \forall (u,v) \in E, w(u) > 0, w(v) < 0 \tag{9}$$

$$2 \cdot x_v \leq \sum_{u \in \delta(v)} x_u \qquad \forall v \in V, w(v) < 0 \tag{10}$$

$$x_v \leq y_v + \sum_{u \in \delta(v)} x_u \qquad \forall v \in V \tag{11}$$

In (7) we require the root node to have a strictly positive weight. We use symmetry breaking constraints (8) to force the node with the smallest index to be the root node. Constraints (9) state that a negatively-weighted node can only be in the solution if all its adjacent positively-weighted nodes are in the solution. In addition, the presence of a node with negative weight in the solution implies that at least two of its neighbors must be in the solution, which is modeled by constraints (10). Constraints (11) are implied by (4) in the case that $|S| = 1$. Adding these constraints results in a tighter upper bound in the initial node of the branch-and-bound tree.

Rooted. The rooted formulation is as follows.

$$\max \sum_{v \in V} w_v x_v \tag{12}$$

$$x_r = 1 \qquad \forall r \in R \tag{13}$$

$$x_v \leq \sum_{u \in \delta(S)} x_u \qquad \forall r \in R, v \in V \setminus R, \{v\} \subseteq S \subseteq V \setminus \{r\} \tag{14}$$

$$x_v \in \{0,1\} \qquad \forall v \in V \tag{15}$$

Constraints (13) enforce the presence of root nodes in the solution. The cut constraints (14) state that a node $v \in V \setminus R$ can only be in the solution if for any root $r \in R$ and for all supersets $S \subseteq V \setminus \{r\}$ of v it holds that a node in the set $\delta(S)$ is in the solution.

We strengthen the formulation using the following cuts.

$$x_v \leq x_u \qquad \forall (u,v) \in E, w(u) > 0, w(v) < 0 \tag{16}$$

$$x_v \leq \sum_{u \in \delta(v)} x_u \qquad \forall v \in V \setminus R \tag{17}$$

Constraints (16) are the same as constraints (9) for the unrooted case. Similarly to the unrooted formulation, constraints (17) correspond to manually adding cuts for the case that $|S| = 1$ in (14).

4.2 Separation

Unrooted. Similarly to [1], the separation problem in the unrooted formulation corresponds to a minimum cut problem on an auxiliary directed support graph D defined as follows: each node $v \in V$ corresponds to an arc (v_1, v_2), and each edge $(u, v) \in E$ corresponds to two arcs (u_2, v_1) and (v_2, u_1). In addition, an artificial root node r is introduced as well as arcs (r, v_1) for all $v \in V$. Given a fractional solution $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$, the arc capacities c are set as follows: $c(r, v_1) = \bar{y}_v$, $c(v_1, v_2) = \bar{x}_v$ and $c(v_2, u_1) = 1$ for all distinct $u, v \in V$. Given a node $v \in V$, we identify violated constraints by solving a minimum cut problem from r to v_2. Let C be a minimum cut set from r to v_2. In case the cut value $c(C)$ is smaller than $\bar{x}_v$, the cut set will admit a set S and $\delta(S)$ such that $\bar{x}_v > \bar{x}(\delta(S)) + \bar{y}(S) = c(C)$. We add such violated constraints to the formulation and resolve again.

Rooted. For the rooted formulation the auxiliary graph D is defined as follows: each node $v \in V \setminus R$ corresponds to an arc (v_1, v_2), and each edge $(u, v) \in E$ corresponds to two arcs (u_2, v_1) and (v_2, u_1) if both u and v not in R. For each root node $r \in R$, a single node is introduced in D. Edges (r, v) incident to a root node $r \in R$ where $v \notin R$ correspond to an arc (r, v_1). We identify violated constraints by identifying minimum cuts between r and v_2 for all $r \in R$ and $v \in V \setminus \{r\}$.

4.3 Primal heuristic

As stated in Section 1, MWCS is solvable in polynomial time for graphs of bounded treewidth. In fact, for trees R-MWCS is solvable in linear time by first rooting the tree at a node $r \in R$ and then solving a dynamic program based on the recurrence:

$$M(v) = w(v) + \sum_{u \in \delta^+(v) \setminus R} \max\{M(u), 0\} + \sum_{u \in \delta^+(v) \cap R} M(u),$$

where $\delta^+(u)$ are the children of the node u.

Our primal heuristic transforms the input graph into a tree by considering the fractional values $\bar{\mathbf{x}}$ given by the solution of the LP relaxation. We use these values to assign an edge cost $c(u, v) = 2 - (\bar{x}_u + \bar{x}_v)$ for each edge $(u, v) \in E$. Next, we compute a minimum-cost spanning tree using Kruskal's algorithm [16]. In the unrooted MWCS case, we root the spanning tree at every positively-weighted node r and assign the solution with maximum weight to be the primal solution. This leads to running time $O(|V|^2)$. In the R-MWCS case, we only root the spanning tree once at an arbitrary vertex $r \in R$, resulting in running time $O(|V|)$.

4.4 Implementation details

Since CPLEX version 12.3, there is a distinction between the user cut callback and the lazy constraint callback. The latter is only called for integral solutions, see Figure 5. Separation of (4) in the case of integral $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ can be done by considering the connected components of the induced subgraph $G[\bar{\mathbf{x}}]$. Let r be the root node encoded in $\bar{\mathbf{y}}$. Recall that (2) ensures that there is only one root node. A connected component C of $G[\bar{\mathbf{x}}]$ that does not contain r corresponds to a violated constraint with $S := C$ and $\delta(S) := \delta(C)$. Violated constraints for R-MWCS in the case of integrality can be separated analogously.

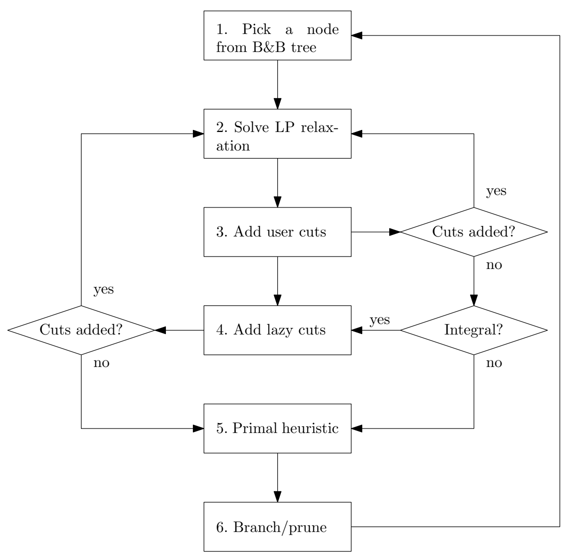

Fig. 5. CPLEX flow diagram.

As can be seen in Figure 5, CPLEX calls the user cut callback at every considered node in the branch-and-bound tree. To prevent spending too much time in the separation and to allow more time for branching, we choose not to separate violated constraints at every callback invocation. Instead we make use of a linear back-off function with an initial waiting period of 1. Upon a successful attempt, the waiting period is incremented by one, thereby gradually decreasing the time spent in separating violated constraints.

5 Results on DIMACS Benchmark

We implemented our algorithm in C++ using the LEMON graph library [7], the OGDF library [6] for building the SPQR tree and the CPLEX v12.6 library for implementing the branch-and-cut approach. Our software tool is called Heinz 2.0

and is available for download at `http://software.cwi.nl/heinz`. The code of the Heinz 2.0 software is managed using github and publicly available under the MIT license at `https://github.com/ls-cwi/heinz`.

We ran all computational experiments on a 12 core Linux machine with a 2.26 GHz Intel Xeon Processor L5640 and 24 GB of RAM, using 2 threads per instance. We used all MWCS instances from the 11th DIMACS Implementation Challenge (`http://dimacs11.cs.princeton.edu`). These are the `ACTMOD` set of 8 instances from integrative network analysis in systems biology and the `JMP_ALM` set of 72 instances, which are based on the random Euclidean instances introduced in [15]. We also considered prize-collecting Steiner tree instances from the DIMACS benchmark, transforming them to MWCS instances using the rule given in Section 1. These are `JMP` (34 instances), `CRR` (80), `PUCNU` (18), `i640` (100), `H` (14), `H2` (14) and `RANDOM` (68). In total we ran computational experiments on 408 instances coming from different applications.

We ran three versions of Heinz 2.0: (i) A pure branch-and-cut approach without preprocessing, to establish a baseline, (ii) preprocessing followed by branch-and-cut, to evaluate the effects of data reduction and (iii) the divide-and-conquer scheme described in Section 3, to evaluate the benefits of the results described in this paper. To allow for a fair comparison, we report only results on instances for which all three methods found feasible solutions. This resulted in 271 instances. A full table of results for all these instances is in the appendix.

For each instance we recorded its size in terms of number of nodes and edges, before and after preprocessing, the best upper and lower bounds that could be found by each of the three methods within a time limit of 6 hours wall time, the running time in wall time, as well as the number of processed biconnected and triconnected components for the divide-and-conquer scheme.

Figure 6 shows the effect of preprocessing. We can observe that preprocessing is effective, reducing more than half of the instances to at most 84% of their original size. Some instances can even be solved by preprocessing. Figure 7 shows the distribution of the optimality gap for the different version of Heinz 2.0. It can be seen that while some instances are hard to solve, both preprocessing and the novel divide-and-conquer scheme provide significant improvements. Also, it can be seen that the PCST instances are harder than the MWCS instances for which all three methods achieve a median gap of 0% Figure 8 shows the distribution of the running times of the instances that were solved to optimality by all three methods. We can see that the divide-and-conquer scheme (median running time of 0.5 s) is faster than the branch-and-cut approach without preprocessing (median running time of 16.4 s). On the MWCS instances, the branch-and-cut approach with processing achieves the same median running time of 0.4 s as the divide-and-conquer scheme. For the PCST instances, however, the divide-and-conquer scheme has the lowest median running time (3.3 s). Moreover, the number of instances that were solved to optimality is the highest for the divide-and-conquer scheme (134), followed by the branch-and-cut approach with preprocessing (129) and the branch-and-cut approach without preprocessing (97).

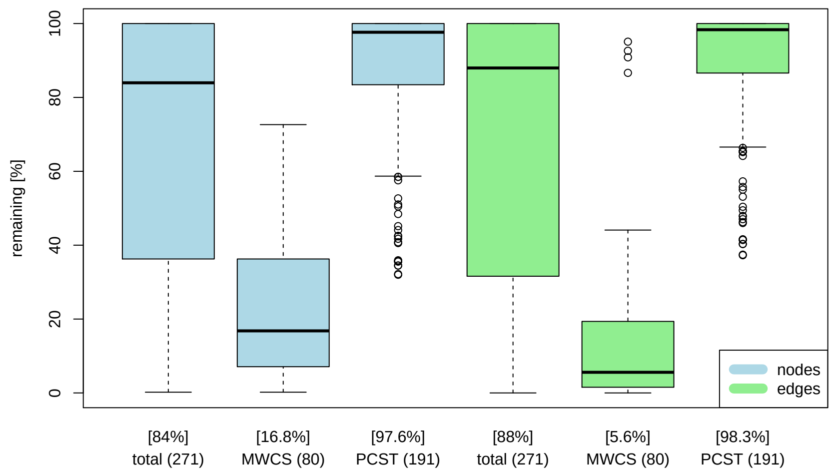

Fig. 6. Effect of preprocessing. The boxplots show the reduction in number of nodes and edges after preprocessing as a fraction of the original value for the 271 instances. The median value is shown in between square brackets, and the number of instances is in between parentheses.

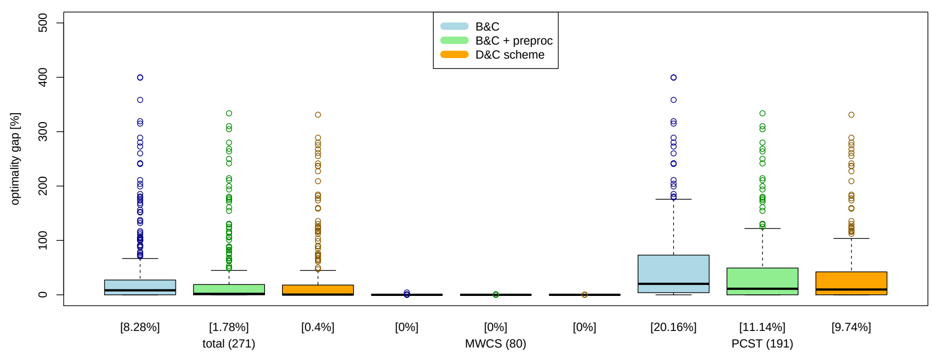

Fig. 7. Distribution of gaps. Boxplots of optimality gap for the three different variants of Heinz 2.0. The median value is shown in between square brackets, and the number of instances is in between parentheses.

6 Conclusions

We have presented a divide-and-conquer scheme for solving the maximum-weight connected subgraph problem to provable optimality. The scheme combines effective preprocessing with a novel decomposition approach that divides an instance into biconnected and triconnected components and solves the core pieces of an instance using branch-and-cut. We have demonstrated the performance of our scheme on the benchmark instances of the 11th DIMACS Implementation Challenge.

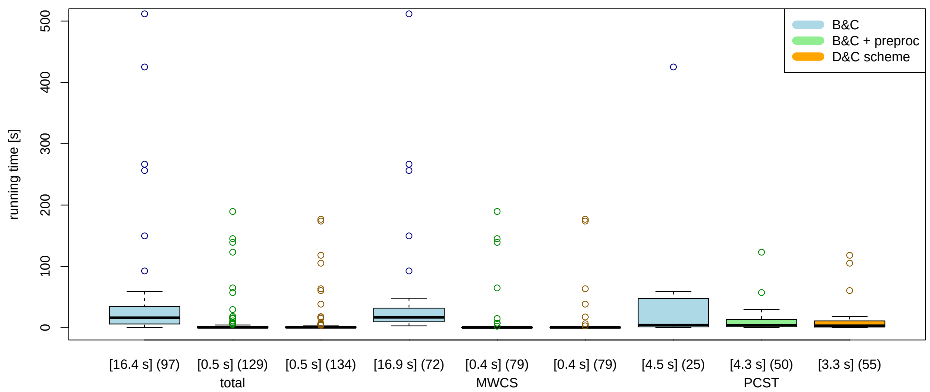

Fig. 8. Distribution of running times. Boxplots of running time (s) of the instances solved to optimality by all three methods within the time limit. The median value is shown in between square brackets, and for each method the total number of instances that it solved to optimality is in between parentheses.

The scheme is modular and allows for the integration of new preprocessing rules or alternative exact algorithms to solve the core instances. We plan, for example, to evaluate a branch-and-cut approach based on an edge-based ILP formulation, which is similar to the one we used for the prize-collecting Steiner tree problem in [18]. Also, we plan to implement an FPT algorithm that can be plugged into the scheme. The modularity of our approach will make it possible to perform extensive algorithm engineering studies and to improve upon the results presented in this paper.

We also want to stress that the divide-and-conquer approach is not specific to MWCS, but also applicable to other types of Steiner problems in graphs. Vice versa, techniques that have been proven useful for related problems may be beneficial for solving MWCS, and we will evaluate their integration into our scheme.

Acknowledgments We thank the participants of the March 2014 NII Shonan Meeting *Towards the ground truth: Exact algorithms for bioinformatics research* and, in particular, Christian Komusiewicz and Falk Hüffner for helpful comments.

References

1. Álvarez-Miranda, E., Ljubic, I., Mutzel, P.: The maximum weight connected subgraph problem. In: Jünger, M., Reinelt, G. (eds.) Facets of Combinatorial Optimization, pp. 245–270. Springer (2013)
2. Backes, C., Rurainski, A., Klau, G., Müller, O., Stöckel, D., Gerasch, A., Küntzer, J., Maisel, D., Ludwig, N., Hein, M., Keller, A., Burtscher, H., Kaufmann, M., Meese, E., Lenhof, H.P.: An integer linear programming approach for finding deregulated subgraphs in regulatory networks. Nucleic Acids Research 40(6), e43 (2012)

3. Bateni, M., Chekuri, C., Ene, A., Hajiaghayi, M.T., Korula, N., Marx, D.: Prize-collecting Steiner problems on planar graphs. In: Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1028–1049 (2011)
4. Carvajal, R., Constantino, M., Goycoolea, M., Pablo Vielma, J., Weintraub, A.: Imposing connectivity constraints in forest planning models. Operations Research 61(4), 824–836 (2013)
5. Chen, C.Y., Grauman, K.: Efficient activity detection with max-subgraph search. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1274–1281. IEEE (2012)
6. Chimani, M., Gutwenger, C., Jünger, M., Klau, G.W., Klein, K., Mutzel, P.: The open graph drawing framework (OGDF). In: Tamassia, R. (ed.) Handbook of Graph Drawing and Visualization. CRC (2013)
7. Dezső, B., Jüttner, A., Kovács, P.: LEMON–an open source C++ graph template library. Electronic Notes in Theoretical Computer Science 264(5), 23–45 (2011)
8. Dilkina, B., Gomes, C.P.: Solving connected subgraph problems in wildlife conservation. In: CPAIOR'10: Proceedings of the 7th international conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Springer (Jun 2010)
9. Dittrich, M.T., Klau, G.W., Rosenwald, A., Dandekar, T., Müller, T.: Identifying functional modules in protein-protein interaction networks: an integrated exact approach. Bioinformatics (Oxford, England) 24(13), i223–31 (Jul 2008)
10. Duin, C.W., Volgenant, A.: Some generalizations of the Steiner problem in graphs. Networks 17(3), 353–364 (1987)
11. Feigenbaum, J., Papadimitriou, C., Shenker, S.: Sharing the cost of muliticast transmissions. In: The thirty-second annual ACM symposium. pp. 218–227. ACM Press, New York, New York, USA (2000)
12. Hochbaum, D.S., Pathria, A.: Node-optimal connected k-subgraphs (1994), uC Berkeley
13. H˝˘ffner, F., Betzler, N., Niedermeier, R.: Separator-based data reduction for signed graph balancing. Journal of Combinatorial Optimization 20(4), 335–360 (2010), http://dx.doi.org/10.1007/s10878-009-9212-2
14. Johnson, D.S.: The NP-completeness column: An ongoing guide. Journal of Algorithms 6(1), 145–159 (Mar 1985)
15. Johnson, D.S., Minkoff, M., Phillips, S.: The prize collecting Steiner tree problem: theory and practice. In: SODA '00: Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics (Feb 2000)
16. Kruskal, J.B.: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proceedings of the American Mathematical Society 7(1), 48 (Feb 1956)
17. Lee, H.F., Dooly, D.R.: Decomposition algorithms for the maximum-weight connected graph problem - Lee - 1999 - Naval Research Logistics (NRL) - Wiley Online Library. Naval Research Logistics (NRL) (1998)
18. Ljubic, I., Weiskircher, R., Pferschy, U., Klau, G.W., Mutzel, P., Fischetti, M.: An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. Mathematical Programming 105(2-3), 427–449 (2006)
19. Yamamoto, T., Bannai, H., Nagasaki, M., Miyano, S.: Better Decomposition Heuristics for the Maximum-Weight Connected Graph Problem Using Betweenness Centrality. In: Discovery Science, pp. 465–472. Springer, Berlin, Heidelberg (2009)

A Detailed Results

The following table lists the results of the instances for which all three methods found feasible solutions. The divide-and-conquer scheme is abbreviated as 'dc', the branch-and-cut approach with preprocessing is 'no-dc' and the branch-and-cut approach without preprocessing is 'no-pre'. The time is in seconds. For results by method 'dc', the last three columns correspond to (from left-to-right) the number of considered blocks, the number of considered triconnected components with at least one nonnegative node and the number of considered triconnected components that only contain negative nodes. For 'no-dc', the last three columns correspond to number of nodes after preprocessing, number of edges after preprocessing and number of components after preprocessing. For results by 'no-pre', the last three columns are empty.

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
ACTMOD	HCMV-7e-4	dc	17.5605	7.55431	7.55431	0.00%	3863	29293	78	11	3	9
ACTMOD	HCMV-7e-4	no-dc	14.9839	7.55431	7.55431	0.00%	3863	29293	78	1387	8188	4
ACTMOD	HCMV-7e-4	no-pre	24.794	7.55431	7.55431	0.00%	3863	29293	78			
ACTMOD	Lymphoma	dc	6.45399	70.1663	70.1663	0.00%	2034	7756	1	1		0
ACTMOD	Lymphoma	no-dc	7.60205	70.1663	70.1663	0.00%	2034	7756	1	1308	6721	1
ACTMOD	Lymphoma	no-pre	2.99873	70.1663	70.1663	0.00%	2034	7756	1			
ACTMOD	metabol-expr-mice-1	dc	21600	547.182	544.948	0.41%	3523	4345	166	1	38	1
ACTMOD	metabol-expr-mice-1	no-dc	21600	550.358	544.948	0.99%	3523	4345	166	666	1069	1
ACTMOD	metabol-expr-mice-1	no-pre	21607.7	567.717	544.249	4.31%	3523	4345	166			
ACTMOD	metabol-expr-mice-2	dc	3.04129	241.078	241.078	0.00%	3514	4332	166	1	18	1
ACTMOD	metabol-expr-mice-2	no-dc	7.21084	241.078	241.078	0.00%	3514	4332	166	637	1029	1
ACTMOD	metabol-expr-mice-2	no-pre	8.97265	241.078	241.078	0.00%	3514	4332	166			
ACTMOD	metabol-expr-mice-3	dc	38.4441	508.302	508.261	0.01%	2853	3335	166	1	23	0
ACTMOD	metabol-expr-mice-3	no-dc	145.228	508.31	508.261	0.01%	2853	3335	166	426	722	1
ACTMOD	metabol-expr-mice-3	no-pre	511.779	508.311	508.261	0.01%	2853	3335	166			
ACTMOD	SSA001	dc	176.86	24.3855	24.3855	0.00%	5226	93394	1	21	2	16
ACTMOD	SSA001	no-dc	139.162	24.3855	24.3855	0.00%	5226	93394	1	3796	88810	1
ACTMOD	SSA001	no-pre	92.6068	24.3855	24.3855	0.00%	5226	93394	1			
ACTMOD	SSA005	dc	173.831	178.664	178.664	0.00%	5226	93394	1	21	5	16
ACTMOD	SSA005	no-dc	189.497	178.675	178.664	0.01%	5226	93394	1	3743	86505	1
ACTMOD	SSA005	no-pre	266.634	178.664	178.664	0.00%	5226	93394	1			
ACTMOD	SSA0075	dc	63.5522	260.524	260.524	0.00%	5226	93394	1	21	7	14
ACTMOD	SSA0075	no-dc	64.9241	260.524	260.524	0.00%	5226	93394	1	3702	84854	1
ACTMOD	SSA0075	no-pre	149.756	260.524	260.524	0.00%	5226	93394	1			
H	hc10	dc	21607.8	243	112	116.96%	6144	10240	1	1		0
H	hc10	no-dc	21608.6	243	108	125.00%	6144	10240	1	6144	10240	1
H	hc10	no-pre	21614.9	243	112	116.96%	6144	10240	1			
H	hc6p	dc	21601.1	1650.25	869	89.90%	256	384	1	1		0
H	hc6p	no-dc	21605	1648.5	882	86.90%	256	384	1	256	384	1
H	hc6p	no-pre	21615.5	1665.56	818	103.61%	256	384	1			
H	hc6u	dc	21600.6	15	10	50.00%	256	384	1	1		0
H	hc6u	no-dc	21615.3	15	10	50.00%	256	384	1	256	384	1
H	hc6u	no-pre	21609.5	15	8	87.50%	256	384	1			
H	hc7p	dc	21599.9	3020.5	1350	123.74%	576	896	1	1		0
H	hc7p	no-dc	21600.2	3020.38	1317	129.34%	576	896	1	576	896	1
H	hc7p	no-pre	21606.3	3030.12	1127	168.87%	576	896	1			
H	hc7u	dc	21599.8	30	17	76.47%	576	896	1	1		0
H	hc7u	no-dc	21599.7	30	18	66.67%	576	896	1	576	896	1

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
H	hc7u	no-pre	21624.8	30	14	114.29%	576	896	1			
H	hc8p	dc	21599.8	6079.29	2353	158.36%	1280	2048	1	1		0
H	hc8p	no-dc	21599.3	6079.91	2390	154.39%	1280	2048	1	1280	2048	1
H	hc8p	no-pre	21636.3	6083.17	2225	173.40%	1280	2048	1			
H	hc8u	dc	21599.6	68	36	88.89%	1280	2048	1	1		0
H	hc8u	no-dc	21599.5	68	36	88.89%	1280	2048	1	1280	2048	1
H	hc8u	no-pre	21648.7	68	33	106.06%	1280	2048	1			
H2	hc10	dc	21602	179	76	135.53%	6144	10240	1	1		0
H2	hc10	no-dc	21602.2	179	83	115.66%	6144	10240	1	6144	10240	1
H2	hc10	no-pre	21612.1	179	76	135.53%	6144	10240	1			
H2	hc6p	dc	21601.1	1650	870	89.66%	256	384	1	1		0
H2	hc6p	no-dc	21604.9	1648.5	877	87.97%	256	384	1	256	384	1
H2	hc6p	no-pre	21608.7	1664.13	826	101.47%	256	384	1			
H2	hc6u	dc	21608	7.5	6	25.00%	256	384	1	1		0
H2	hc6u	no-dc	21608.5	7.5	6	25.00%	256	384	1	256	384	1
H2	hc6u	no-pre	21605.3	10	6	66.67%	256	384	1			
H2	hc7p	dc	21600.2	3703.83	2001	85.10%	576	896	1	1		0
H2	hc7p	no-dc	21599.6	3706.24	2027	82.84%	576	896	1	576	896	1
H2	hc7p	no-pre	21619.1	3711.67	1855	100.09%	576	896	1			
H2	hc7u	dc	21599.8	23	14	64.29%	576	896	1	1		0
H2	hc7u	no-dc	21600.4	23	15	53.33%	576	896	1	576	896	1
H2	hc7u	no-pre	21622.5	23	12	91.67%	576	896	1			
H2	hc8p	dc	21599.8	7383.37	3627	103.57%	1280	2048	1	1		0
H2	hc8p	no-dc	21599.8	7383.67	3576	106.48%	1280	2048	1	1280	2048	1
H2	hc8p	no-pre	21653.9	7391.58	3186	132.00%	1280	2048	1			
H2	hc8u	dc	21599.5	41	19	115.79%	1280	2048	1	1		0
H2	hc8u	no-dc	21599.6	41	19	115.79%	1280	2048	1	1280	2048	1
H2	hc8u	no-pre	21636.2	41	16	156.25%	1280	2048	1			
H2	hc9u	dc	21600.2	83	38	118.42%	2816	4608	1	1		0
H2	hc9u	no-dc	21599.7	83	36	130.56%	2816	4608	1	2816	4608	1
H2	hc9u	no-pre	21658.7	83	33	151.52%	2816	4608	1			
i640	i640-001-PCST	dc	96.5367	1524	1524	0.00%	1600	1920	1	1	3	0
i640	i640-001-PCST	no-dc	93.8082	1524	1524	0.00%	1600	1920	1	956	1274	1
i640	i640-001-PCST	no-pre	21599.5	1682.23	1524	10.38%	1600	1920	1			
i640	i640-002-PCST	dc	105.124	801	801	0.00%	1600	1920	1	1	4	0
i640	i640-002-PCST	no-dc	224.695	801	801	0.00%	1600	1920	1	921	1232	1
i640	i640-002-PCST	no-pre	21599.6	907.412	801	13.28%	1600	1920	1			
i640	i640-003-PCST	dc	377.454	914	914	0.00%	1600	1920	1	1	2	0
i640	i640-003-PCST	no-dc	279.681	914	914	0.00%	1600	1920	1	936	1254	1
i640	i640-003-PCST	no-pre	21599.6	1063.41	914	16.35%	1600	1920	1			
i640	i640-004-PCST	dc	2518.64	1668	1668	0.00%	1600	1920	1	1	7	0
i640	i640-004-PCST	no-dc	2690.31	1668	1668	0.00%	1600	1920	1	936	1252	2
i640	i640-004-PCST	no-pre	21608.1	1954.61	1579	23.79%	1600	1920	1			
i640	i640-005-PCST	dc	105.225	820	820	0.00%	1600	1920	1	1	8	0
i640	i640-005-PCST	no-dc	29.6425	820	820	0.00%	1600	1920	1	968	1286	2
i640	i640-005-PCST	no-pre	58.8116	820	820	0.00%	1600	1920	1			
i640	i640-011-PCST	dc	21605.3	2121	1938	9.44%	4775	8270	1	1		0
i640	i640-011-PCST	no-dc	21604	2120.53	1938	9.42%	4775	8270	1	4775	8270	1
i640	i640-011-PCST	no-pre	21612.2	2285.24	1877	21.75%	4775	8270	1			
i640	i640-012-PCST	dc	21609.8	3401.04	3245	4.81%	4775	8270	1	1		0
i640	i640-012-PCST	no-dc	21608.8	3393.5	3245	4.58%	4775	8270	1	4775	8270	1
i640	i640-012-PCST	no-pre	21600.5	3552.42	3138	13.21%	4775	8270	1			
i640	i640-013-PCST	dc	21599.9	2163.75	2066	4.73%	4775	8270	1	1		0
i640	i640-013-PCST	no-dc	21600	2160.88	2066	4.59%	4775	8270	1	4775	8270	1
i640	i640-013-PCST	no-pre	21603.7	2331.88	2056	13.42%	4775	8270	1			
i640	i640-014-PCST	dc	21599.9	4657.45	4553	2.29%	4775	8270	1	1		0

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
i640	i640-014-PCST	no-dc	21599.9	4668.88	4553	2.55%	4775	8270	1	4775	8270	1
i640	i640-014-PCST	no-pre	21603.6	4866.29	4467	8.94%	4775	8270	1			
i640	i640-015-PCST	dc	21600.2	3559.1	3353	6.15%	4775	8270	1	1		0
i640	i640-015-PCST	no-dc	21600.1	3559.25	3353	6.15%	4775	8270	1	4775	8270	1
i640	i640-015-PCST	no-pre	21607.1	3743.05	3306	13.22%	4775	8270	1			
i640	i640-021-PCST	dc	21600.5	4792.5	4011	19.48%	205120	408960	1	1		0
i640	i640-021-PCST	no-dc	21600.4	4792.48	4009	19.54%	205120	408960	1	205120	408960	1
i640	i640-021-PCST	no-pre	21607.7	4792.5	3919	22.29%	205120	408960	1			
i640	i640-022-PCST	dc	21602.8	3801.5	3207	18.54%	205120	408960	1	1		0
i640	i640-022-PCST	no-dc	21600.9	3729	3263	14.28%	205120	408960	1	205120	408960	1
i640	i640-022-PCST	no-pre	21621.5	3842.5	3154	21.83%	205120	408960	1			
i640	i640-024-PCST	dc	21600.4	4170.09	3606	15.64%	205120	408960	1	1		0
i640	i640-024-PCST	no-dc	21600.6	4249	3602	17.96%	205120	408960	1	205120	408960	1
i640	i640-024-PCST	no-pre	21629.8	4180.63	3498	19.51%	205120	408960	1			
i640	i640-031-PCST	dc	91.4961	1044	1044	0.00%	1920	2560	1	1		1
i640	i640-031-PCST	no-dc	102.739	1044	1044	0.00%	1920	2560	1	1617	2256	2
i640	i640-031-PCST	no-pre	21602.5	1124.98	1044	7.76%	1920	2560	1			
i640	i640-032-PCST	dc	51.4425	1223	1223	0.00%	1920	2560	1	1	3	0
i640	i640-032-PCST	no-dc	45.2618	1223	1223	0.00%	1920	2560	1	1596	2236	1
i640	i640-032-PCST	no-pre	21599.4	1341.83	1223	9.72%	1920	2560	1			
i640	i640-033-PCST	dc	18039.1	819	819	0.00%	1920	2560	1	1	2	0
i640	i640-033-PCST	no-dc	14695.9	819	819	0.00%	1920	2560	1	1609	2248	2
i640	i640-033-PCST	no-pre	21603.3	1112.24	797	39.55%	1920	2560	1			
i640	i640-034-PCST	dc	569.678	2211	2211	0.00%	1920	2560	1	1	3	2
i640	i640-034-PCST	no-dc	941.881	2211	2211	0.00%	1920	2560	1	1612	2252	1
i640	i640-034-PCST	no-pre	21604.4	2472.47	2134	15.86%	1920	2560	1			
i640	i640-035-PCST	dc	21599.4	1338.6	1286	4.09%	1920	2560	1	1	1	0
i640	i640-035-PCST	no-dc	21599.4	1343.84	1286	4.50%	1920	2560	1	1604	2244	1
i640	i640-035-PCST	no-pre	21599.6	1587	1286	23.41%	1920	2560	1			
i640	i640-041-PCST	dc	21599.7	5501	5238	5.02%	41536	81792	1	1		0
i640	i640-041-PCST	no-dc	21599.8	5498.25	5255	4.63%	41536	81792	1	41536	81792	1
i640	i640-041-PCST	no-pre	21623.5	5569.71	4972	12.02%	41536	81792	1			
i640	i640-042-PCST	dc	21600	4023.5	3758	7.06%	41536	81792	1	1		0
i640	i640-042-PCST	no-dc	21599.3	4025	3726	8.02%	41536	81792	1	41536	81792	1
i640	i640-042-PCST	no-pre	21601.6	4086.2	3617	12.97%	41536	81792	1			
i640	i640-043-PCST	dc	21599.7	2096.06	1907	9.91%	41536	81792	1	1		0
i640	i640-043-PCST	no-dc	21599.6	2083.25	1907	9.24%	41536	81792	1	41536	81792	1
i640	i640-043-PCST	no-pre	21624.5	2170.99	1902	14.14%	41536	81792	1			
i640	i640-044-PCST	dc	21599.9	4216.5	3892	8.34%	41536	81792	1	1		0
i640	i640-044-PCST	no-dc	21599.9	4223	3892	8.50%	41536	81792	1	41536	81792	1
i640	i640-044-PCST	no-pre	21622.9	4279.65	3775	13.37%	41536	81792	1			
i640	i640-045-PCST	dc	21599.7	3731.83	3509	6.35%	41536	81792	1	1		0
i640	i640-045-PCST	no-dc	21599.7	3744.56	3509	6.71%	41536	81792	1	41536	81792	1
i640	i640-045-PCST	no-pre	21601.9	3788.14	3499	8.26%	41536	81792	1			
i640	i640-101-PCST	dc	21600.5	6127.64	5440	12.64%	1600	1920	1	1	9	0
i640	i640-101-PCST	no-dc	21600.5	6020.5	5437	10.73%	1600	1920	1	970	1286	1
i640	i640-101-PCST	no-pre	21609.7	6334.38	5159	22.78%	1600	1920	1			
i640	i640-102-PCST	dc	21599.8	7089.44	6821	3.94%	1600	1920	1	1	16	0
i640	i640-102-PCST	no-dc	21600.4	7120.11	6821	4.39%	1600	1920	1	961	1278	1
i640	i640-102-PCST	no-pre	21606.3	7362.75	6800	8.28%	1600	1920	1			
i640	i640-103-PCST	dc	21600.2	6884.07	6526	5.49%	1600	1920	1	1	7	0
i640	i640-103-PCST	no-dc	21600.7	6854.67	6498	5.49%	1600	1920	1	939	1256	1
i640	i640-103-PCST	no-pre	21607	7132.05	6413	11.21%	1600	1920	1			
i640	i640-104-PCST	dc	21599.6	3248.78	3083	5.38%	1600	1920	1	1	15	0
i640	i640-104-PCST	no-dc	21600.1	3294.29	3079	6.99%	1600	1920	1	940	1256	1
i640	i640-104-PCST	no-pre	21604.2	3742.48	2955	26.65%	1600	1920	1			

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
i640	i640-105-PCST	dc	21599.8	8160.4	7243	12.67%	1600	1920	1	1	13	0
i640	i640-105-PCST	no-dc	21600.2	8108.46	7262	11.66%	1600	1920	1	981	1300	1
i640	i640-105-PCST	no-pre	21611.7	8327.74	6953	19.77%	1600	1920	1			
i640	i640-112-PCST	dc	21600	9486.13	8622	10.02%	4775	8270	1	1		0
i640	i640-112-PCST	no-dc	21599.7	9497.22	8664	9.62%	4775	8270	1	4775	8270	1
i640	i640-112-PCST	no-pre	21624.7	9616.36	8200	17.27%	4775	8270	1			
i640	i640-113-PCST	dc	21600.3	9438.31	8252	14.38%	4775	8270	1	1		0
i640	i640-113-PCST	no-dc	21599.6	9416.05	8223	14.51%	4775	8270	1	4775	8270	1
i640	i640-113-PCST	no-pre	21600.9	9563.7	7959	20.16%	4775	8270	1			
i640	i640-114-PCST	dc	21600.2	8484.92	7502	13.10%	4775	8270	1	1		0
i640	i640-114-PCST	no-dc	21599.5	8485.92	7486	13.36%	4775	8270	1	4775	8270	1
i640	i640-114-PCST	no-pre	21624.4	8596.27	6930	24.04%	4775	8270	1			
i640	i640-115-PCST	dc	21600.1	10337.3	9018	14.63%	4775	8270	1	1		0
i640	i640-115-PCST	no-dc	21599.7	10346.5	9057	14.24%	4775	8270	1	4775	8270	1
i640	i640-115-PCST	no-pre	21625.8	10451.4	8493	23.06%	4775	8270	1			
i640	i640-131-PCST	dc	21599.9	6596.19	6090	8.31%	1920	2560	1	1	11	2
i640	i640-131-PCST	no-dc	21599.9	6579	6089	8.05%	1920	2560	1	1612	2252	1
i640	i640-131-PCST	no-pre	21615.7	6783.43	5783	17.30%	1920	2560	1			
i640	i640-132-PCST	dc	21599.6	9922.09	9122	8.77%	1920	2560	1	1	7	1
i640	i640-132-PCST	no-dc	21600.4	9909.74	9122	8.64%	1920	2560	1	1611	2250	2
i640	i640-132-PCST	no-pre	21609.6	10228.1	8816	16.02%	1920	2560	1			
i640	i640-133-PCST	dc	21600	9772.67	9016	8.39%	1920	2560	1	1	6	1
i640	i640-133-PCST	no-dc	21600.3	9792.83	9008	8.71%	1920	2560	1	1614	2254	1
i640	i640-133-PCST	no-pre	21616.1	9989.11	8788	13.67%	1920	2560	1			
i640	i640-134-PCST	dc	21599.6	7695	7240	6.28%	1920	2560	1	1	7	0
i640	i640-134-PCST	no-dc	21599.6	7720.32	7245	6.56%	1920	2560	1	1622	2262	1
i640	i640-134-PCST	no-pre	21599.9	7919.95	7063	12.13%	1920	2560	1			
i640	i640-135-PCST	dc	21599.8	10667	10083	5.79%	1920	2560	1	1	8	0
i640	i640-135-PCST	no-dc	21599.7	10700	10076	6.19%	1920	2560	1	1610	2250	1
i640	i640-135-PCST	no-pre	21611.4	10946.8	9548	14.65%	1920	2560	1			
i640	i640-141-PCST	dc	21600.3	15567.8	13708	13.57%	41536	81792	1	1		0
i640	i640-141-PCST	no-dc	21604.2	15614.5	13759	13.49%	41536	81792	1	41536	81792	1
i640	i640-141-PCST	no-pre	21637.2	15751.3	12943	21.70%	41536	81792	1			
i640	i640-142-PCST	dc	21600.1	10871	9740	11.61%	41536	81792	1	1		0
i640	i640-142-PCST	no-dc	21600.3	10873.5	9775	11.24%	41536	81792	1	41536	81792	1
i640	i640-142-PCST	no-pre	21655.8	11063.3	8958	23.50%	41536	81792	1			
i640	i640-144-PCST	dc	21601	9257.5	8218	12.65%	41536	81792	1	1		0
i640	i640-144-PCST	no-dc	21600.3	9260.25	8174	13.29%	41536	81792	1	41536	81792	1
i640	i640-144-PCST	no-pre	21636.1	9468.9	7365	28.57%	41536	81792	1			
i640	i640-145-PCST	dc	21599.7	11742	10193	15.20%	41536	81792	1	1		0
i640	i640-145-PCST	no-dc	21599.4	11758.8	10186	15.44%	41536	81792	1	41536	81792	1
i640	i640-145-PCST	no-pre	21633.2	11816	9717	21.60%	41536	81792	1			
i640	i640-201-PCST	dc	21600.3	14582.1	13352	9.21%	1600	1920	1	1	25	0
i640	i640-201-PCST	no-dc	21600.6	14719.7	13235	11.22%	1600	1920	1	979	1298	1
i640	i640-201-PCST	no-pre	21619.6	15135.2	13144	15.15%	1600	1920	1			
i640	i640-202-PCST	dc	21600	16725.7	15281	9.45%	1600	1920	1	1	30	0
i640	i640-202-PCST	no-dc	21600.3	16817.5	15387	9.30%	1600	1920	1	973	1288	2
i640	i640-202-PCST	no-pre	21615.9	17170	14739	16.49%	1600	1920	1			
i640	i640-203-PCST	dc	21599.5	16355.5	14906	9.72%	1600	1920	1	1	29	0
i640	i640-203-PCST	no-dc	21600	16423.2	15056	9.08%	1600	1920	1	984	1302	1
i640	i640-203-PCST	no-pre	21618.5	16616.9	14090	17.93%	1600	1920	1			
i640	i640-204-PCST	dc	21599.8	18342	16934	8.31%	1600	1920	1	1	33	0
i640	i640-204-PCST	no-dc	21600	18389.2	16971	8.36%	1600	1920	1	968	1282	1
i640	i640-204-PCST	no-pre	21607.1	18763.5	16236	15.57%	1600	1920	1			
i640	i640-205-PCST	dc	21599.8	18099	16496	9.72%	1600	1920	1	1	30	0
i640	i640-205-PCST	no-dc	21599.8	18330.8	16494	11.14%	1600	1920	1	992	1308	1

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
i640	i640-205-PCST	no-pre	21621.7	18510	16430	12.66%	1600	1920	1			
i640	i640-211-PCST	dc	21600.1	23170.8	19830	16.85%	4775	8270	1	1		0
i640	i640-211-PCST	no-dc	21599.5	23168	20031	15.66%	4775	8270	1	4775	8270	1
i640	i640-211-PCST	no-pre	21638.6	23330.1	18980	22.92%	4775	8270	1			
i640	i640-212-PCST	dc	21600.2	21178.5	17924	18.16%	4775	8270	1	1		0
i640	i640-212-PCST	no-dc	21599.6	21193.2	17870	18.60%	4775	8270	1	4775	8270	1
i640	i640-212-PCST	no-pre	21619.3	21260.3	17548	21.16%	4775	8270	1			
i640	i640-213-PCST	dc	21600.1	20649	17245	19.74%	4775	8270	1	1		0
i640	i640-213-PCST	no-dc	21599.7	20609.4	17486	17.86%	4775	8270	1	4775	8270	1
i640	i640-213-PCST	no-pre	21630.5	20701.5	16625	24.52%	4775	8270	1			
i640	i640-214-PCST	dc	21600	20757.6	17457	18.91%	4775	8270	1	1		0
i640	i640-214-PCST	no-dc	21599.4	20794.2	17458	19.11%	4775	8270	1	4775	8270	1
i640	i640-214-PCST	no-pre	21611.6	20844.3	17070	22.11%	4775	8270	1			
i640	i640-215-PCST	dc	21599.9	22305.6	19061	17.02%	4775	8270	1	1		0
i640	i640-215-PCST	no-dc	21599.6	22285.4	19068	16.87%	4775	8270	1	4775	8270	1
i640	i640-215-PCST	no-pre	21641.1	22359.1	18062	23.79%	4775	8270	1			
i640	i640-231-PCST	dc	21599.5	19934.4	17286	15.32%	1920	2560	1	1	12	0
i640	i640-231-PCST	no-dc	21599.5	20045	17445	14.90%	1920	2560	1	1626	2266	1
i640	i640-231-PCST	no-pre	21628.2	20268.8	16337	24.07%	1920	2560	1			
i640	i640-232-PCST	dc	21600	18039.8	15735	14.65%	1920	2560	1	1	7	0
i640	i640-232-PCST	no-dc	21599.5	18066.1	15634	15.56%	1920	2560	1	1630	2270	1
i640	i640-232-PCST	no-pre	21629.4	18344.1	15376	19.30%	1920	2560	1			
i640	i640-233-PCST	dc	21599.6	16069.1	13691	17.37%	1920	2560	1	1	15	1
i640	i640-233-PCST	no-dc	21599.4	16299.4	13786	18.23%	1920	2560	1	1668	2308	1
i640	i640-233-PCST	no-pre	21619.6	16392	13570	20.80%	1920	2560	1			
i640	i640-234-PCST	dc	21599.8	20622.2	18682	10.39%	1920	2560	1	1	19	4
i640	i640-234-PCST	no-dc	21599.6	20718.4	18781	10.32%	1920	2560	1	1632	2272	1
i640	i640-234-PCST	no-pre	21634.3	21008.5	17874	17.54%	1920	2560	1			
i640	i640-235-PCST	dc	21600.3	15816	13819	14.45%	1920	2560	1	1	11	0
i640	i640-235-PCST	no-dc	21599.5	15849.3	13840	14.52%	1920	2560	1	1608	2248	1
i640	i640-235-PCST	no-pre	21600.3	16072.6	13315	20.71%	1920	2560	1			
i640	i640-243-PCST	dc	21601.4	22726.5	18106	25.52%	41536	81792	1	1		0
i640	i640-243-PCST	no-dc	21602.2	22741	17916	26.93%	41536	81792	1	41536	81792	1
i640	i640-243-PCST	no-pre	21701.8	22747.1	17416	30.61%	41536	81792	1			
i640	i640-245-PCST	dc	21612.8	24486.2	19290	26.94%	41536	81792	1	1		0
i640	i640-245-PCST	no-dc	21600.4	24605.6	19465	26.41%	41536	81792	1	41536	81792	1
i640	i640-245-PCST	no-pre	21629.6	24628	18726	31.52%	41536	81792	1			
i640	i640-301-PCST	dc	21599.7	64707.3	57638	12.26%	1600	1920	1	1	71	0
i640	i640-301-PCST	no-dc	21600	66342	57712	14.95%	1600	1920	1	1073	1392	1
i640	i640-301-PCST	no-pre	21602.3	66771.9	56520	18.14%	1600	1920	1			
i640	i640-302-PCST	dc	21599.7	59345.1	52302	13.47%	1600	1920	1	1	75	0
i640	i640-302-PCST	no-dc	21600.2	60489.7	51612	17.20%	1600	1920	1	1031	1350	1
i640	i640-302-PCST	no-pre	21634	60152.4	51395	17.04%	1600	1920	1			
i640	i640-303-PCST	dc	21600.1	67578.5	61529	9.83%	1600	1920	1	1	109	0
i640	i640-303-PCST	no-dc	21600.3	68628.7	60657	13.14%	1600	1920	1	1076	1394	1
i640	i640-303-PCST	no-pre	21634.4	68969.5	60128	14.70%	1600	1920	1			
i640	i640-304-PCST	dc	21599.7	57784.3	50978	13.35%	1600	1920	1	1	66	0
i640	i640-304-PCST	no-dc	21606.9	58896.8	50115	17.52%	1600	1920	1	1060	1376	1
i640	i640-304-PCST	no-pre	21633.7	58747.7	49525	18.62%	1600	1920	1			
i640	i640-305-PCST	dc	21599.8	70376.4	63267	11.24%	1600	1920	1	1	82	0
i640	i640-305-PCST	no-dc	21600.3	71237.6	62518	13.95%	1600	1920	1	1043	1362	1
i640	i640-305-PCST	no-pre	21612.4	71426.4	61803	15.57%	1600	1920	1			
i640	i640-312-PCST	dc	21599.7	76496.5	62083	23.22%	4775	8270	1	1		0
i640	i640-312-PCST	no-dc	21599.4	76591.3	62477	22.59%	4775	8270	1	4773	8268	1
i640	i640-312-PCST	no-pre	21658.3	77053.2	60233	27.93%	4775	8270	1			
i640	i640-331-PCST	dc	21600.1	69226.8	60082	15.22%	1920	2560	1	1	44	0

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
i640	i640-331-PCST	no-dc	21618.7	70025.4	59450	17.79%	1920	2560	1	1652	2292	1
i640	i640-331-PCST	no-pre	21660.7	70279.1	57910	21.36%	1920	2560	1			
i640	i640-334-PCST	dc	21600.2	68613.8	58285	17.72%	1920	2560	1	1	40	1
i640	i640-334-PCST	no-dc	21599.7	68944.2	57884	19.11%	1920	2560	1	1668	2308	1
i640	i640-334-PCST	no-pre	21644	69066.5	57132	20.89%	1920	2560	1			
i640	i640-335-PCST	dc	21600.3	66682.2	56638	17.73%	1920	2560	1	1	54	1
i640	i640-335-PCST	no-dc	21599.6	67352.1	56224	19.79%	1920	2560	1	1682	2322	1
i640	i640-335-PCST	no-pre	21642.8	67440.4	54445	23.87%	1920	2560	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.25-e-0.25	dc	0.35959	931.539	931.539	0.00%	1000	4936	1	2	1	7
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.25-e-0.25	no-dc	0.376951	931.539	931.539	0.00%	1000	4936	1	551	2147	1
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.25-e-0.25	no-pre	21642.5	932.208	927.748	0.48%	1000	4936	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.25-e-0.5	dc	0.443036	1872.28	1872.28	0.00%	1000	4936	1	12		8
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.25-e-0.5	no-dc	0.413477	1872.28	1872.28	0.00%	1000	4936	1	308	905	1
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.25-e-0.5	no-pre	14.6974	1872.28	1872.28	0.00%	1000	4936	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.25-e-0.75	dc	0.688218	2789.58	2789.58	0.00%	1000	4936	1	20		0
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.25-e-0.75	no-dc	2.51025	2789.58	2789.58	0.00%	1000	4936	1	89	206	1
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.25-e-0.75	no-pre	16.2762	2789.58	2789.58	0.00%	1000	4936	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.5-e-0.25	dc	1.03506	522.526	522.526	0.00%	1000	4936	1	9		5
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.5-e-0.25	no-dc	1.11962	522.526	522.526	0.00%	1000	4936	1	348	994	1
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.5-e-0.25	no-pre	17.0304	522.526	522.526	0.00%	1000	4936	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.5-e-0.5	dc	11.2558	1197.85	1197.85	0.00%	1000	4936	1	18		2
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.5-e-0.5	no-dc	0.188367	1197.85	1197.85	0.00%	1000	4936	1	129	317	1
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.5-e-0.5	no-pre	21615.9	1197.85	1187.86	0.84%	1000	4936	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.5-e-0.75	dc	0.166488	1762.71	1762.71	0.00%	1000	4936	1	3		0
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.5-e-0.75	no-dc	0.14509	1762.71	1762.71	0.00%	1000	4936	1	10	18	1
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.5-e-0.75	no-pre	21618.2	1762.71	1754.66	0.46%	1000	4936	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.75-e-0.25	dc	0.597751	332.792	332.792	0.00%	1000	4936	1	15		0
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.75-e-0.25	no-dc	0.513384	332.792	332.792	0.00%	1000	4936	1	79	190	1
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.75-e-0.25	no-pre	19.4802	332.792	332.792	0.00%	1000	4936	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.75-e-0.5	dc	1.35743	754.301	754.301	0.00%	1000	4936	1	8		0
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.75-e-0.5	no-dc	0.202057	754.301	754.301	0.00%	1000	4936	1	28	56	1
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.75-e-0.5	no-pre	22.1738	754.301	754.301	0.00%	1000	4936	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.75-e-0.75	dc	0.262642	998.215	998.215	0.00%	1000	4936	1	1		0
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.75-e-0.75	no-dc	0.254381	998.215	998.215	0.00%	1000	4936	1	4	6	1
JMP-ALM-K	MWCS-I-D-n-1000-a-0.6-d-0.75-e-0.75	no-pre	24.3359	998.215	998.215	0.00%	1000	4936	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.25-e-0.25	dc	0.571053	939.393	939.393	0.00%	1000	13279	1	1		0
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.25-e-0.25	no-dc	0.362946	939.393	939.393	0.00%	1000	13279	1	559	4680	1
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.25-e-0.25	no-pre	16.1951	939.393	939.393	0.00%	1000	13279	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.25-e-0.5	dc	0.599238	1883.21	1883.21	0.00%	1000	13279	1	2		0
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.25-e-0.5	no-dc	0.519799	1883.21	1883.21	0.00%	1000	13279	1	361	2069	1
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.25-e-0.5	no-pre	21.7189	1883.21	1883.21	0.00%	1000	13279	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.25-e-0.75	dc	0.479617	2789.58	2789.58	0.00%	1000	13279	1	1		4
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.25-e-0.75	no-dc	0.464305	2789.58	2789.58	0.00%	1000	13279	1	176	621	1
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.25-e-0.75	no-pre	22.9324	2789.58	2789.58	0.00%	1000	13279	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.5-e-0.25	dc	0.774964	533.429	533.429	0.00%	1000	13279	1	1		0
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.5-e-0.25	no-dc	0.533593	533.429	533.429	0.00%	1000	13279	1	380	2286	1
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.5-e-0.25	no-pre	24.9116	533.429	533.429	0.00%	1000	13279	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.5-e-0.5	dc	0.509635	1205.42	1205.42	0.00%	1000	13279	1	1		2
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.5-e-0.5	no-dc	0.61424	1205.42	1205.42	0.00%	1000	13279	1	242	987	1
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.5-e-0.5	no-pre	21.101	1205.42	1205.42	0.00%	1000	13279	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.5-e-0.75	dc	0.526392	1770.28	1770.28	0.00%	1000	13279	1	7		2
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.5-e-0.75	no-dc	0.458529	1770.28	1770.28	0.00%	1000	13279	1	87	228	1
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.5-e-0.75	no-pre	23.6105	1770.28	1770.28	0.00%	1000	13279	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.75-e-0.25	dc	0.451476	336.83	336.83	0.00%	1000	13279	1	3		4
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.75-e-0.25	no-dc	0.49937	336.83	336.83	0.00%	1000	13279	1	171	631	1
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.75-e-0.25	no-pre	21.9377	336.83	336.83	0.00%	1000	13279	1			

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.75-e-0.5	dc	0.48971	760.285	760.285	0.00%	1000	13279	1	11		0
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.75-e-0.5	no-dc	0.399399	760.285	760.285	0.00%	1000	13279	1	81	219	1
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.75-e-0.5	no-pre	24.3109	760.285	760.285	0.00%	1000	13279	1			
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.75-e-0.75	dc	0.455312	1004.2	1004.2	0.00%	1000	13279	1	5		0
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.75-e-0.75	no-dc	0.395662	1004.2	1004.2	0.00%	1000	13279	1	19	41	1
JMP-ALM-K	MWCS-I-D-n-1000-a-1-d-0.75-e-0.75	no-pre	25.7161	1004.2	1004.2	0.00%	1000	13279	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.25-e-0.25	dc	1.86749	1333.52	1333.4	0.01%	1500	7662	1	5	1	11
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.25-e-0.25	no-dc	1.74175	1333.53	1333.48	0.00%	1500	7662	1	843	3367	1
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.25-e-0.25	no-pre	21620.4	1335.9	1332.91	0.22%	1500	7662	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.25-e-0.5	dc	0.570849	2799.68	2799.68	0.00%	1500	7662	1	16		3
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.25-e-0.5	no-dc	0.602353	2799.68	2799.68	0.00%	1500	7662	1	459	1381	1
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.25-e-0.5	no-pre	48.0926	2799.68	2799.68	0.00%	1500	7662	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.25-e-0.75	dc	0.495041	4230.25	4230.25	0.00%	1500	7662	1	22		0
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.25-e-0.75	no-dc	0.386206	4230.25	4230.25	0.00%	1500	7662	1	108	261	1
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.25-e-0.75	no-pre	43.8777	4230.25	4230.25	0.00%	1500	7662	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.5-e-0.25	dc	0.527668	847.452	847.452	0.00%	1500	7662	1	15		17
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.5-e-0.25	no-dc	0.530441	847.452	847.452	0.00%	1500	7662	1	506	1562	1
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.5-e-0.25	no-pre	256.341	847.452	847.452	0.00%	1500	7662	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.5-e-0.5	dc	0.514145	1858.09	1858.09	0.00%	1500	7662	1	31		2
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.5-e-0.5	no-dc	0.377694	1858.09	1858.09	0.00%	1500	7662	1	217	544	1
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.5-e-0.5	no-pre	21601.2	1858.09	1852.25	0.32%	1500	7662	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.5-e-0.75	dc	0.466714	2697.46	2697.46	0.00%	1500	7662	1	12		0
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.5-e-0.75	no-dc	0.388014	2697.46	2697.46	0.00%	1500	7662	1	41	85	1
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.5-e-0.75	no-pre	46.716	2697.46	2697.46	0.00%	1500	7662	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.75-e-0.25	dc	0.62229	502.176	502.176	0.00%	1500	7662	1	22		0
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.75-e-0.25	no-dc	0.481436	502.176	502.176	0.00%	1500	7662	1	103	244	1
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.75-e-0.25	no-pre	31.2525	502.176	502.176	0.00%	1500	7662	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.75-e-0.5	dc	0.439236	1089.77	1089.77	0.00%	1500	7662	1	10		0
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.75-e-0.5	no-dc	0.38177	1089.77	1089.77	0.00%	1500	7662	1	34	71	1
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.75-e-0.5	no-pre	34.7852	1089.77	1089.77	0.00%	1500	7662	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.75-e-0.75	dc	0.412308	1423.61	1423.61	0.00%	1500	7662	1	2		0
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.75-e-0.75	no-dc	0.39237	1423.61	1423.61	0.00%	1500	7662	1	7	12	1
JMP-ALM-K	MWCS-I-D-n-1500-a-0.6-d-0.75-e-0.75	no-pre	45.4047	1423.61	1423.61	0.00%	1500	7662	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.25-e-0.25	dc	0.950446	1377.01	1377.01	0.00%	1500	20527	1	1		0
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.25-e-0.25	no-dc	0.84676	1377.01	1377.01	0.00%	1500	20527	1	836	7246	1
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.25-e-0.25	no-pre	28.0418	1377.01	1377.01	0.00%	1500	20527	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.25-e-0.5	dc	1.03289	2820.05	2820.05	0.00%	1500	20527	1	1		3
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.25-e-0.5	no-dc	0.714584	2820.05	2820.05	0.00%	1500	20527	1	544	3241	1
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.25-e-0.5	no-pre	32.6055	2820.05	2820.05	0.00%	1500	20527	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.25-e-0.75	dc	1.12183	4230.25	4230.25	0.00%	1500	20527	1	8		0
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.25-e-0.75	no-dc	1.0126	4230.25	4230.25	0.00%	1500	20527	1	252	918	1
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.25-e-0.75	no-pre	45.0324	4230.25	4230.25	0.00%	1500	20527	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.5-e-0.25	dc	0.910596	860.619	860.619	0.00%	1500	20527	1	1		0
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.5-e-0.25	no-dc	0.781462	860.619	860.619	0.00%	1500	20527	1	568	3494	1
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.5-e-0.25	no-pre	30.2049	860.619	860.619	0.00%	1500	20527	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.5-e-0.5	dc	0.817512	1865.66	1865.66	0.00%	1500	20527	1	1		0
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.5-e-0.5	no-dc	1.04705	1865.66	1865.66	0.00%	1500	20527	1	364	1546	1
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.5-e-0.5	no-pre	39.2171	1865.66	1865.66	0.00%	1500	20527	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.5-e-0.75	dc	0.94015	2707.7	2707.7	0.00%	1500	20527	1	9		2
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.5-e-0.75	no-dc	0.891467	2707.7	2707.7	0.00%	1500	20527	1	158	435	1
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.5-e-0.75	no-pre	43.5297	2707.7	2707.7	0.00%	1500	20527	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.75-e-0.25	dc	1.17345	502.176	502.176	0.00%	1500	20527	1	4		3
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.75-e-0.25	no-dc	0.855199	502.176	502.176	0.00%	1500	20527	1	235	828	1
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.75-e-0.25	no-pre	21601	502.176	497.471	0.95%	1500	20527	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.75-e-0.5	dc	0.977376	1089.77	1089.77	0.00%	1500	20527	1	9		3
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.75-e-0.5	no-dc	0.849739	1089.77	1089.77	0.00%	1500	20527	1	105	277	1

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.75-e-0.5	no-pre	40.0845	1089.77	1089.77	0.00%	1500	20527	1			
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.75-e-0.75	dc	0.979065	1423.61	1423.61	0.00%	1500	20527	1	4		0
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.75-e-0.75	no-dc	0.908807	1423.61	1423.61	0.00%	1500	20527	1	15	30	1
JMP-ALM-K	MWCS-I-D-n-1500-a-1-d-0.75-e-0.75	no-pre	38.4752	1423.61	1423.61	0.00%	1500	20527	1			
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.25-e-0.25	dc	0.337464	460.577	460.577	0.00%	500	2597	1	1		4
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.25-e-0.25	no-dc	0.33266	460.577	460.577	0.00%	500	2597	1	284	1145	1
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.25-e-0.25	no-pre	21608.7	461.426	459.025	0.52%	500	2597	1			
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.25-e-0.5	dc	0.339386	992.967	992.967	0.00%	500	2597	1	6		1
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.25-e-0.5	no-dc	0.345827	992.967	992.967	0.00%	500	2597	1	149	431	1
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.25-e-0.5	no-pre	3.38871	992.967	992.967	0.00%	500	2597	1			
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.25-e-0.75	dc	0.332261	1447.54	1447.54	0.00%	500	2597	1	11		0
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.25-e-0.75	no-dc	0.412452	1447.54	1447.54	0.00%	500	2597	1	44	100	1
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.25-e-0.75	no-pre	4.19754	1447.54	1447.54	0.00%	500	2597	1			
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.5-e-0.25	dc	0.187674	280.832	280.832	0.00%	500	2597	1	5		3
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.5-e-0.25	no-dc	0.168414	280.832	280.832	0.00%	500	2597	1	183	578	1
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.5-e-0.25	no-pre	5.41741	280.832	280.832	0.00%	500	2597	1			
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.5-e-0.5	dc	0.200553	655.623	655.623	0.00%	500	2597	1	9		0
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.5-e-0.5	no-dc	0.142848	655.623	655.623	0.00%	500	2597	1	77	199	1
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.5-e-0.5	no-pre	4.8355	655.623	655.623	0.00%	500	2597	1			
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.5-e-0.75	dc	0.178045	965.555	965.555	0.00%	500	2597	1	3		0
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.5-e-0.75	no-dc	0.121274	965.555	965.555	0.00%	500	2597	1	10	18	1
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.5-e-0.75	no-pre	5.16469	965.555	965.555	0.00%	500	2597	1			
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.75-e-0.25	dc	0.241102	171.629	171.629	0.00%	500	2597	1	10		0
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.75-e-0.25	no-dc	0.13804	171.629	171.629	0.00%	500	2597	1	38	82	1
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.75-e-0.25	no-pre	4.74162	171.629	171.629	0.00%	500	2597	1			
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.75-e-0.5	dc	0.16164	362.188	362.188	0.00%	500	2597	1	3		0
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.75-e-0.5	no-dc	0.147642	362.188	362.188	0.00%	500	2597	1	10	18	1
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.75-e-0.5	no-pre	5.00884	362.188	362.188	0.00%	500	2597	1			
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.75-e-0.75	dc	0.150595	490.624	490.624	0.00%	500	2597	1			0
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.75-e-0.75	no-dc	0.0974782	490.624	490.624	0.00%	500	2597	1	1	0	1
JMP-ALM-K	MWCS-I-D-n-500-a-0.62-d-0.75-e-0.75	no-pre	5.97095	490.624	490.624	0.00%	500	2597	1			
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.25-e-0.25	dc	0.14857	471.393	471.393	0.00%	500	6519	1	1		0
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.25-e-0.25	no-dc	0.109316	471.393	471.393	0.00%	500	6519	1	290	2427	1
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.25-e-0.25	no-pre	3.52505	471.393	471.393	0.00%	500	6519	1			
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.25-e-0.5	dc	0.239974	995.313	995.313	0.00%	500	6519	1	1		0
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.25-e-0.5	no-dc	0.220264	995.313	995.313	0.00%	500	6519	1	184	992	1
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.25-e-0.5	no-pre	6.20482	995.313	995.313	0.00%	500	6519	1			
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.25-e-0.75	dc	0.183088	1447.54	1447.54	0.00%	500	6519	1	1		3
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.25-e-0.75	no-dc	0.190553	1447.54	1447.54	0.00%	500	6519	1	87	285	1
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.25-e-0.75	no-pre	8.55337	1447.54	1447.54	0.00%	500	6519	1			
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.5-e-0.25	dc	0.228956	286.921	286.921	0.00%	500	6519	1	1		0
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.5-e-0.25	no-dc	0.314466	286.921	286.921	0.00%	500	6519	1	196	1212	1
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.5-e-0.25	no-pre	4.64682	286.921	286.921	0.00%	500	6519	1			
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.5-e-0.5	dc	0.209118	661.712	661.712	0.00%	500	6519	1	1		3
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.5-e-0.5	no-dc	0.218419	661.712	661.712	0.00%	500	6519	1	119	497	1
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.5-e-0.5	no-pre	7.64028	661.712	661.712	0.00%	500	6519	1			
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.5-e-0.75	dc	0.183704	965.555	965.555	0.00%	500	6519	1	5		2
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.5-e-0.75	no-dc	0.208071	965.555	965.555	0.00%	500	6519	1	35	90	1
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.5-e-0.75	no-pre	10.1341	965.555	965.555	0.00%	500	6519	1			
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.75-e-0.25	dc	0.243299	171.629	171.629	0.00%	500	6519	1	4		2
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.75-e-0.25	no-dc	0.288803	171.629	171.629	0.00%	500	6519	1	81	262	1
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.75-e-0.25	no-pre	7.46269	171.629	171.629	0.00%	500	6519	1			
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.75-e-0.5	dc	0.105109	362.188	362.188	0.00%	500	6519	1	4		3
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.75-e-0.5	no-dc	0.120824	362.188	362.188	0.00%	500	6519	1	39	101	1
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.75-e-0.5	no-pre	10.1377	362.188	362.188	0.00%	500	6519	1			
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.75-e-0.75	dc	0.198998	490.624	490.624	0.00%	500	6519	1	2		0

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.75-e-0.75	no-dc	0.18676	490.624	490.624	0.00%	500	6519	1	7	12	1
JMP-ALM-K	MWCS-I-D-n-500-a-1-d-0.75-e-0.75	no-pre	11.2697	490.624	490.624	0.00%	500	6519	1			
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.25-e-0.25	dc	0.225135	702.644	702.644	0.00%	750	4219	1	3		2
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.25-e-0.25	no-dc	0.212163	702.644	702.644	0.00%	750	4219	1	417	1748	1
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.25-e-0.25	no-pre	10270	702.644	702.644	0.00%	750	4219	1			
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.25-e-0.5	dc	0.231107	1419.78	1419.78	0.00%	750	4219	1	7		3
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.25-e-0.5	no-dc	0.366497	1419.78	1419.78	0.00%	750	4219	1	233	699	1
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.25-e-0.5	no-pre	11598.1	1419.78	1419.78	0.00%	750	4219	1			
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.25-e-0.75	dc	0.245836	2116.58	2116.58	0.00%	750	4219	1	14		0
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.25-e-0.75	no-dc	0.198132	2116.58	2116.58	0.00%	750	4219	1	73	168	1
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.25-e-0.75	no-pre	11.0224	2116.58	2116.58	0.00%	750	4219	1			
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.5-e-0.25	dc	0.201506	403.178	403.178	0.00%	750	4219	1	2		6
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.5-e-0.25	no-dc	0.17438	403.178	403.178	0.00%	750	4219	1	274	875	1
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.5-e-0.25	no-pre	9.09528	403.178	403.178	0.00%	750	4219	1			
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.5-e-0.5	dc	0.267254	946.129	946.129	0.00%	750	4219	1	13		2
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.5-e-0.5	no-dc	0.283627	946.129	946.129	0.00%	750	4219	1	126	319	1
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.5-e-0.5	no-pre	12.4868	946.129	946.129	0.00%	750	4219	1			
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.5-e-0.75	dc	0.224284	1382.77	1382.77	0.00%	750	4219	1	6		0
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.5-e-0.75	no-dc	0.252632	1382.77	1382.77	0.00%	750	4219	1	19	36	1
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.5-e-0.75	no-pre	13.4736	1382.77	1382.77	0.00%	750	4219	1			
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.75-e-0.25	dc	0.266241	266.984	266.984	0.00%	750	4219	1	11		2
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.75-e-0.25	no-dc	0.254452	266.984	266.984	0.00%	750	4219	1	67	172	1
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.75-e-0.25	no-pre	11.1406	266.984	266.984	0.00%	750	4219	1			
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.75-e-0.5	dc	0.21649	580.408	580.408	0.00%	750	4219	1	5		0
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.75-e-0.5	no-dc	0.186797	580.408	580.408	0.00%	750	4219	1	19	41	1
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.75-e-0.5	no-pre	14.9693	580.408	580.408	0.00%	750	4219	1			
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.75-e-0.75	dc	0.222651	764.157	764.157	0.00%	750	4219	1	2		0
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.75-e-0.75	no-dc	0.218307	764.157	764.157	0.00%	750	4219	1	7	12	1
JMP-ALM-K	MWCS-I-D-n-750-a-0.647-d-0.75-e-0.75	no-pre	17.1973	764.157	764.157	0.00%	750	4219	1			
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.25-e-0.25	dc	0.485725	708.144	708.144	0.00%	750	9822	1	1		0
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.25-e-0.25	no-dc	0.387681	708.144	708.144	0.00%	750	9822	1	422	3477	1
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.25-e-0.25	no-pre	7.55873	708.144	708.144	0.00%	750	9822	1			
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.25-e-0.5	dc	0.396535	1426.45	1426.45	0.00%	750	9822	1	1		2
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.25-e-0.5	no-dc	0.368243	1426.45	1426.45	0.00%	750	9822	1	272	1464	1
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.25-e-0.5	no-pre	14.4751	1426.45	1426.45	0.00%	750	9822	1			
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.25-e-0.75	dc	1.03195	2116.58	2116.58	0.00%	750	9822	1	1		5
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.25-e-0.75	no-dc	0.918836	2116.58	2116.58	0.00%	750	9822	1	131	426	1
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.25-e-0.75	no-pre	15.5529	2116.58	2116.58	0.00%	750	9822	1			
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.5-e-0.25	dc	0.416411	403.178	403.178	0.00%	750	9822	1	1		0
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.5-e-0.25	no-dc	0.306639	403.178	403.178	0.00%	750	9822	1	294	1821	1
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.5-e-0.25	no-pre	11.0283	403.178	403.178	0.00%	750	9822	1			
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.5-e-0.5	dc	0.357499	946.129	946.129	0.00%	750	9822	1	1		0
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.5-e-0.5	no-dc	0.403572	946.129	946.129	0.00%	750	9822	1	185	761	1
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.5-e-0.5	no-pre	16.4334	946.129	946.129	0.00%	750	9822	1			
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.5-e-0.75	dc	0.351872	1382.77	1382.77	0.00%	750	9822	1	8		2
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.5-e-0.75	no-dc	0.291939	1382.77	1382.77	0.00%	750	9822	1	62	151	1
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.5-e-0.75	no-pre	16.6906	1382.77	1382.77	0.00%	750	9822	1			
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.75-e-0.25	dc	0.321332	266.984	266.984	0.00%	750	9822	1	3		3
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.75-e-0.25	no-dc	0.42309	266.984	266.984	0.00%	750	9822	1	125	439	1
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.75-e-0.25	no-pre	15.3708	266.984	266.984	0.00%	750	9822	1			
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.75-e-0.5	dc	0.4494	580.408	580.408	0.00%	750	9822	1	6		3
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.75-e-0.5	no-dc	0.322315	580.408	580.408	0.00%	750	9822	1	49	130	1
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.75-e-0.5	no-pre	16.7571	580.408	580.408	0.00%	750	9822	1			
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.75-e-0.75	dc	0.443031	764.157	764.157	0.00%	750	9822	1	3		0
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.75-e-0.75	no-dc	0.334512	764.157	764.157	0.00%	750	9822	1	11	22	1
JMP-ALM-K	MWCS-I-D-n-750-a-1-d-0.75-e-0.75	no-pre	19.9133	764.157	764.157	0.00%	750	9822	1			

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
PCSPG-CRR	C01-A	dc	0.395731	9	9	0.00%	1125	1250	1	1	1	0
PCSPG-CRR	C01-A	no-dc	0.308369	9	9	0.00%	1125	1250	1	402	518	2
PCSPG-CRR	C01-A	no-pre	0.350259	12	9	33.33%	1125	1250	1			
PCSPG-CRR	C01-B	dc	2.26586	189	189	0.00%	1125	1250	1	1	2	0
PCSPG-CRR	C01-B	no-dc	2.00749	189	189	0.00%	1125	1250	1	404	520	2
PCSPG-CRR	C01-B	no-pre	1202.02	189	189	0.00%	1125	1250	1			
PCSPG-CRR	C02-A	dc	0.405445	9	9	0.00%	1125	1250	1	1	2	0
PCSPG-CRR	C02-A	no-dc	0.359965	9	9	0.00%	1125	1250	1	360	466	1
PCSPG-CRR	C02-A	no-pre	0.47153	9	9	0.00%	1125	1250	1			
PCSPG-CRR	C02-B	dc	15.2858	463	463	0.00%	1125	1250	1	1	4	0
PCSPG-CRR	C02-B	no-dc	16.1315	463	463	0.00%	1125	1250	1	362	468	1
PCSPG-CRR	C02-B	no-pre	852.486	463	463	0.00%	1125	1250	1			
PCSPG-CRR	C03-A	dc	2.6277	25	25	0.00%	1125	1250	1	1	24	0
PCSPG-CRR	C03-A	no-dc	11.1454	25	25	0.00%	1125	1250	1	459	578	1
PCSPG-CRR	C03-A	no-pre	425.147	25	25	0.00%	1125	1250	1			
PCSPG-CRR	C03-B	dc	21600.2	3782.78	3713	1.88%	1125	1250	1	1	39	0
PCSPG-CRR	C03-B	no-dc	21599.6	3803.8	3712	2.47%	1125	1250	1	477	598	1
PCSPG-CRR	C03-B	no-pre	21608.5	3815.94	3711	2.83%	1125	1250	1			
PCSPG-CRR	C04-A	dc	13882.9	30	30	0.00%	1125	1250	1	1	42	0
PCSPG-CRR	C04-A	no-dc	21600	43.4444	30	44.81%	1125	1250	1	496	618	1
PCSPG-CRR	C04-A	no-pre	21606	52.968	30	76.56%	1125	1250	1			
PCSPG-CRR	C04-B	dc	21600.3	5570.5	5480	1.65%	1125	1250	1	1	64	0
PCSPG-CRR	C04-B	no-dc	21599.6	5592.5	5484	1.98%	1125	1250	1	508	630	1
PCSPG-CRR	C04-B	no-pre	21618.7	5609.32	5472	2.51%	1125	1250	1			
PCSPG-CRR	C05-A	dc	21600.7	218.5	165	32.42%	1125	1250	1	1	75	0
PCSPG-CRR	C05-A	no-dc	21599.5	248.75	152	63.65%	1125	1250	1	545	664	2
PCSPG-CRR	C05-A	no-pre	21617.6	252.455	148	70.58%	1125	1250	1			
PCSPG-CRR	C05-B	dc	21600.6	10919	10889	0.28%	1125	1250	1	1	89	0
PCSPG-CRR	C05-B	no-dc	21599.8	11143.6	10953	1.74%	1125	1250	1	568	688	2
PCSPG-CRR	C05-B	no-pre	21623.7	11156.6	10948	1.91%	1125	1250	1			
PCSPG-CRR	C06-A	dc	1.44116	11.3333	9	25.93%	1500	2000	1	1	1	0
PCSPG-CRR	C06-A	no-dc	1.61292	11.3333	9	25.93%	1500	2000	1	1203	1674	1
PCSPG-CRR	C06-A	no-pre	0.899602	11.3333	9	25.93%	1500	2000	1			
PCSPG-CRR	C06-B	dc	54.3774	219	219	0.00%	1500	2000	1	1	1	0
PCSPG-CRR	C06-B	no-dc	52.975	219	219	0.00%	1500	2000	1	1203	1674	1
PCSPG-CRR	C06-B	no-pre	21601.8	221.107	219	0.96%	1500	2000	1			
PCSPG-CRR	C07-A	dc	5.18326	9	9	0.00%	1500	2000	1	1	4	0
PCSPG-CRR	C07-A	no-dc	4.42913	9	9	0.00%	1500	2000	1	1249	1732	1
PCSPG-CRR	C07-A	no-pre	4.08032	9	9	0.00%	1500	2000	1			
PCSPG-CRR	C07-B	dc	80.7827	502	502	0.00%	1500	2000	1	1	5	0
PCSPG-CRR	C07-B	no-dc	97.0853	502	502	0.00%	1500	2000	1	1249	1732	1
PCSPG-CRR	C07-B	no-pre	21603.7	510	495	3.03%	1500	2000	1			
PCSPG-CRR	C08-A	dc	21600	108.556	75	44.74%	1500	2000	1	1	18	0
PCSPG-CRR	C08-A	no-dc	21600.4	108.306	75	44.41%	1500	2000	1	1255	1736	1
PCSPG-CRR	C08-A	no-pre	21620.3	114.613	71	61.43%	1500	2000	1			
PCSPG-CRR	C08-B	dc	21599.9	4030.48	3940	2.30%	1500	2000	1	1	24	0
PCSPG-CRR	C08-B	no-dc	21599.4	4031.57	3927	2.66%	1500	2000	1	1255	1736	1
PCSPG-CRR	C08-B	no-pre	21625.4	4038.39	3931	2.73%	1500	2000	1			
PCSPG-CRR	C09-A	dc	21606	166.471	97	71.62%	1500	2000	1	1	30	0
PCSPG-CRR	C09-A	no-dc	21603.6	168.234	103	63.33%	1500	2000	1	1321	1808	2
PCSPG-CRR	C09-A	no-pre	21620.8	175.75	92	91.03%	1500	2000	1			
PCSPG-CRR	C09-B	dc	21600.4	5966.86	5817	2.58%	1500	2000	1	1	40	0
PCSPG-CRR	C09-B	no-dc	21600.6	5982.33	5799	3.16%	1500	2000	1	1324	1812	1
PCSPG-CRR	C09-B	no-pre	21611.1	5989.59	5796	3.34%	1500	2000	1			
PCSPG-CRR	C10-A	dc	21599.6	459.875	320	43.71%	1500	2000	1	1	50	0
PCSPG-CRR	C10-A	no-dc	21599.6	494	329	50.15%	1500	2000	1	1341	1826	1

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
PCSPG-CRR	C10-A	no-pre	21649.2	488.139	328	48.82%	1500	2000	1			
PCSPG-CRR	C10-B	dc	21600.6	11561.7	11352	1.85%	1500	2000	1	1	51	0
PCSPG-CRR	C10-B	no-dc	21600.7	11604.7	11335	2.38%	1500	2000	1	1341	1826	1
PCSPG-CRR	C10-B	no-pre	21663.2	11684.5	11274	3.64%	1500	2000	1			
PCSPG-CRR	C11-A	dc	4.74838	9	9	0.00%	3000	5000	1	1		0
PCSPG-CRR	C11-A	no-dc	4.13165	9	9	0.00%	3000	5000	1	3000	5000	1
PCSPG-CRR	C11-A	no-pre	4.14716	9	9	0.00%	3000	5000	1			
PCSPG-CRR	C11-B	dc	798.51	242	242	0.00%	3000	5000	1	1		0
PCSPG-CRR	C11-B	no-dc	873.506	242	242	0.00%	3000	5000	1	3000	5000	1
PCSPG-CRR	C11-B	no-pre	21601.1	247.387	241	2.65%	3000	5000	1			
PCSPG-CRR	C12-A	dc	18.0262	21	21	0.00%	3000	5000	1	1		0
PCSPG-CRR	C12-A	no-dc	18.4456	21	21	0.00%	3000	5000	1	2998	4998	1
PCSPG-CRR	C12-A	no-pre	34.1415	21	21	0.00%	3000	5000	1			
PCSPG-CRR	C12-B	dc	4681.27	558	558	0.00%	3000	5000	1	1		0
PCSPG-CRR	C12-B	no-dc	21599.4	559.495	558	0.27%	3000	5000	1	2998	4998	1
PCSPG-CRR	C12-B	no-pre	21611.6	563.61	557	1.19%	3000	5000	1			
PCSPG-CRR	C13-A	dc	21600	238.75	187	27.67%	3000	5000	1	1		0
PCSPG-CRR	C13-A	no-dc	21604.4	238.7	187	27.65%	3000	5000	1	2998	4998	1
PCSPG-CRR	C13-A	no-pre	21609.9	244.667	175	39.81%	3000	5000	1			
PCSPG-CRR	C13-B	dc	21601.6	4251.22	4171	1.92%	3000	5000	1	1		0
PCSPG-CRR	C13-B	no-dc	21599.5	4250.25	4176	1.78%	3000	5000	1	2998	4998	1
PCSPG-CRR	C13-B	no-pre	21610.6	4255	4161	2.26%	3000	5000	1			
PCSPG-CRR	C14-A	dc	21600.4	408.208	319	27.96%	3000	5000	1	1		0
PCSPG-CRR	C14-A	no-dc	21599.9	408.234	316	29.19%	3000	5000	1	3000	5000	1
PCSPG-CRR	C14-A	no-pre	21691.2	413.083	300	37.69%	3000	5000	1			
PCSPG-CRR	C14-B	dc	21600.2	6312	6179	2.15%	3000	5000	1	1		0
PCSPG-CRR	C14-B	no-dc	21600.3	6313.67	6180	2.16%	3000	5000	1	3000	5000	1
PCSPG-CRR	C14-B	no-pre	21619.1	6318.64	6167	2.46%	3000	5000	1			
PCSPG-CRR	C15-A	dc	21601.9	836.083	652	28.23%	3000	5000	1	1		0
PCSPG-CRR	C15-A	no-dc	21601.3	834.523	677	23.27%	3000	5000	1	3000	5000	1
PCSPG-CRR	C15-A	no-pre	21611.7	840.178	670	25.40%	3000	5000	1			
PCSPG-CRR	C16-A	dc	17152.5	16	16	0.00%	13000	25000	1	1		0
PCSPG-CRR	C16-A	no-dc	12363.9	16	16	0.00%	13000	25000	1	13000	25000	1
PCSPG-CRR	C16-A	no-pre	21620.3	18.443	14	31.74%	13000	25000	1			
PCSPG-CRR	C16-B	dc	11250.2	263	263	0.00%	13000	25000	1	1		0
PCSPG-CRR	C16-B	no-dc	6925.77	263	263	0.00%	13000	25000	1	13000	25000	1
PCSPG-CRR	C16-B	no-pre	21602.2	265.5	260	2.12%	13000	25000	1			
PCSPG-CRR	C17-A	dc	7814.2	41	41	0.00%	13000	25000	1	1		0
PCSPG-CRR	C17-A	no-dc	9473.24	41	41	0.00%	13000	25000	1	13000	25000	1
PCSPG-CRR	C17-A	no-pre	21608.6	43.9303	40	9.83%	13000	25000	1			
PCSPG-CRR	C17-B	dc	15780.4	586	586	0.00%	13000	25000	1	1		0
PCSPG-CRR	C17-B	no-dc	13023.3	586	586	0.00%	13000	25000	1	13000	25000	1
PCSPG-CRR	C17-B	no-pre	21622.8	590	585	0.85%	13000	25000	1			
PCSPG-CRR	D01-A	dc	0.99705	9	9	0.00%	2250	2500	1	1	3	0
PCSPG-CRR	D01-A	no-dc	0.748952	9	9	0.00%	2250	2500	1	776	1008	1
PCSPG-CRR	D01-A	no-pre	0.541995	9	9	0.00%	2250	2500	1			
PCSPG-CRR	D01-B	dc	118.131	168	168	0.00%	2250	2500	1	1	4	0
PCSPG-CRR	D01-B	no-dc	123.133	168	168	0.00%	2250	2500	1	776	1008	1
PCSPG-CRR	D01-B	no-pre	3160.23	168	168	0.00%	2250	2500	1			
PCSPG-CRR	D02-A	dc	0.924299	9	9	0.00%	2250	2500	1	1	7	0
PCSPG-CRR	D02-A	no-dc	0.823907	9	9	0.00%	2250	2500	1	800	1036	1
PCSPG-CRR	D02-A	no-pre	0.869112	9	9	0.00%	2250	2500	1			
PCSPG-CRR	D02-B	dc	27.6653	386	386	0.00%	2250	2500	1	1	7	0
PCSPG-CRR	D02-B	no-dc	88.148	386	386	0.00%	2250	2500	1	802	1038	1
PCSPG-CRR	D02-B	no-pre	21617.4	399.88	383	4.41%	2250	2500	1			
PCSPG-CRR	D03-A	dc	21600.1	49.0833	40	22.71%	2250	2500	1	1	81	0

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
PCSPG-CRR	D03-A	no-dc	21600.5	54.0189	40	35.05%	2250	2500	1	913	1150	1
PCSPG-CRR	D03-A	no-pre	21614.1	69.4929	40	73.73%	2250	2500	1			
PCSPG-CRR	D03-B	dc	21599.6	7064.36	6921	2.07%	2250	2500	1	1	107	0
PCSPG-CRR	D03-B	no-dc	21599.5	7134.59	6930	2.95%	2250	2500	1	939	1176	1
PCSPG-CRR	D03-B	no-pre	21606.4	7149.21	6926	3.22%	2250	2500	1			
PCSPG-CRR	D04-A	dc	21599.6	75.1786	44	70.86%	2250	2500	1	1	74	0
PCSPG-CRR	D04-A	no-dc	21604.3	87.3624	41	113.08%	2250	2500	1	957	1198	1
PCSPG-CRR	D04-A	no-pre	21604.3	98.4246	39	152.37%	2250	2500	1			
PCSPG-CRR	D05-A	dc	21600.1	440.824	300	46.94%	2250	2500	1	1	159	0
PCSPG-CRR	D05-A	no-dc	21600.5	492.656	293	68.14%	2250	2500	1	1148	1394	1
PCSPG-CRR	D05-A	no-pre	21634.4	503.231	281	79.09%	2250	2500	1			
PCSPG-CRR	D05-B	dc	21600.1	21699.8	21562	0.64%	2250	2500	1	1	168	0
PCSPG-CRR	D05-B	no-dc	21601.3	22105.8	21633	2.19%	2250	2500	1	1185	1432	1
PCSPG-CRR	D05-B	no-pre	21635.7	22171.6	21614	2.58%	2250	2500	1			
PCSPG-CRR	D06-A	dc	1.24949	12.6667	9	40.74%	3000	4000	1	1	2	0
PCSPG-CRR	D06-A	no-dc	1.27034	12.6667	9	40.74%	3000	4000	1	2530	3530	1
PCSPG-CRR	D06-A	no-pre	1.22917	9	9	0.00%	3000	4000	1			
PCSPG-CRR	D06-B	dc	3003.68	207	207	0.00%	3000	4000	1	1	2	0
PCSPG-CRR	D06-B	no-dc	3330.83	207	207	0.00%	3000	4000	1	2530	3530	1
PCSPG-CRR	D06-B	no-pre	21602.2	215.333	206	4.53%	3000	4000	1			
PCSPG-CRR	D07-A	dc	6.92653	9	9	0.00%	3000	4000	1	1	2	1
PCSPG-CRR	D07-A	no-dc	2.79269	9	9	0.00%	3000	4000	1	2520	3520	1
PCSPG-CRR	D07-A	no-pre	8.26165	9	9	0.00%	3000	4000	1			
PCSPG-CRR	D07-B	dc	348.517	501	501	0.00%	3000	4000	1	1	2	1
PCSPG-CRR	D07-B	no-dc	275.843	501	501	0.00%	3000	4000	1	2520	3520	1
PCSPG-CRR	D07-B	no-pre	21601.1	509	501	1.60%	3000	4000	1			
PCSPG-CRR	D08-A	dc	21600.3	149.591	86	73.94%	3000	4000	1	1	51	1
PCSPG-CRR	D08-A	no-dc	21779.5	150.661	83	81.52%	3000	4000	1	2606	3606	1
PCSPG-CRR	D08-A	no-pre	21611.4	156.222	74	111.11%	3000	4000	1			
PCSPG-CRR	D08-B	dc	21600.4	7625.05	7378	3.35%	3000	4000	1	1	73	0
PCSPG-CRR	D08-B	no-dc	21667.5	7651.42	7368	3.85%	3000	4000	1	2610	3610	1
PCSPG-CRR	D08-B	no-pre	21673	7643.41	7367	3.75%	3000	4000	1			
PCSPG-CRR	D09-A	dc	21600.5	297.833	133	123.93%	3000	4000	1	1	66	1
PCSPG-CRR	D09-A	no-dc	21723.6	283.353	141	100.96%	3000	4000	1	2619	3616	4
PCSPG-CRR	D09-A	no-pre	21647.4	310.333	123	152.30%	3000	4000	1			
PCSPG-CRR	D09-B	dc	21601.2	11311.2	10964	3.17%	3000	4000	1	1	86	1
PCSPG-CRR	D09-B	no-dc	21713	11335.8	10930	3.71%	3000	4000	1	2616	3616	1
PCSPG-CRR	D09-B	no-pre	21628.6	11394	10924	4.30%	3000	4000	1			
PCSPG-CRR	D11-A	dc	4.03291	9	9	0.00%	6000	10000	1	1		0
PCSPG-CRR	D11-A	no-dc	3.4761	9	9	0.00%	6000	10000	1	5988	9988	1
PCSPG-CRR	D11-A	no-pre	3.35759	9	9	0.00%	6000	10000	1			
PCSPG-CRR	D11-B	dc	2790.42	245	245	0.00%	6000	10000	1	1		0
PCSPG-CRR	D11-B	no-dc	3605.05	245	245	0.00%	6000	10000	1	5988	9988	1
PCSPG-CRR	D11-B	no-pre	21602.8	251.625	243	3.55%	6000	10000	1			
PCSPG-CRR	D12-A	dc	17527.8	17	17	0.00%	6000	10000	1	1		0
PCSPG-CRR	D12-A	no-dc	21599.5	18.0833	17	6.37%	6000	10000	1	6000	10000	1
PCSPG-CRR	D12-A	no-pre	21614	23.0239	16	43.90%	6000	10000	1			
PCSPG-CRR	D12-B	dc	16586	562	562	0.00%	6000	10000	1	1		0
PCSPG-CRR	D12-B	no-dc	19446.1	562	562	0.00%	6000	10000	1	6000	10000	1
PCSPG-CRR	D12-B	no-pre	21619.8	568.159	561	1.28%	6000	10000	1			
PCSPG-CRR	D13-A	dc	21600.6	495.167	307	61.29%	6000	10000	1	1		0
PCSPG-CRR	D13-A	no-dc	21600.3	481.042	324	48.47%	6000	10000	1	5996	9996	1
PCSPG-CRR	D13-A	no-pre	21672.8	496.197	288	72.29%	6000	10000	1			
PCSPG-CRR	D16-A	dc	1266.79	14	14	0.00%	26000	50000	1	1		0
PCSPG-CRR	D16-A	no-dc	1564.22	14	14	0.00%	26000	50000	1	26000	50000	1
PCSPG-CRR	D16-A	no-pre	21603	16.4841	14	17.74%	26000	50000	1			

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
PCSPG-CRR	D16-B	dc	2144.1	261	261	0.00%	26000	50000	1	1		0
PCSPG-CRR	D16-B	no-dc	2650.15	261	261	0.00%	26000	50000	1	26000	50000	1
PCSPG-CRR	D16-B	no-pre	21604.5	264.108	261	1.19%	26000	50000	1			
PCSPG-CRR	D17-A	dc	21601	38.5	36	6.94%	26000	50000	1	1		0
PCSPG-CRR	D17-A	no-dc	21601.1	38.4737	36	6.87%	26000	50000	1	26000	50000	1
PCSPG-CRR	D17-A	no-pre	21622.8	42.1468	33	27.72%	26000	50000	1			
PCSPG-CRR	D17-B	dc	21600.2	583.35	581	0.40%	26000	50000	1	1		0
PCSPG-CRR	D17-B	no-dc	21613.8	583.667	581	0.46%	26000	50000	1	26000	50000	1
PCSPG-CRR	D17-B	no-pre	21625.9	586.378	580	1.10%	26000	50000	1			
PCSPG-JMP	K100	dc	1.28926	26226	26226	0.00%	451	702	3	4		1
PCSPG-JMP	K100	no-dc	1.71095	26226	26226	0.00%	451	702	3	428	670	2
PCSPG-JMP	K100	no-pre	1.78865	26226	26226	0.00%	451	702	3			
PCSPG-JMP	K100.1	dc	1.61789	25222	25222	0.00%	448	696	1	1	1	2
PCSPG-JMP	K100.1	no-dc	1.28505	25222	25222	0.00%	448	696	1	438	682	1
PCSPG-JMP	K100.1	no-pre	1.91519	25222	25222	0.00%	448	696	1			
PCSPG-JMP	K100.10	dc	1.62208	29160	29160	0.00%	419	638	1	4	1	0
PCSPG-JMP	K100.10	no-dc	1.37012	29160	29160	0.00%	419	638	1	408	622	1
PCSPG-JMP	K100.10	no-pre	1.4056	29160	29160	0.00%	419	638	1			
PCSPG-JMP	K100.2	dc	21600.5	44261.3	42985	2.97%	439	678	1	2	1	1
PCSPG-JMP	K100.2	no-dc	21613.4	51066.5	42985	18.80%	439	678	1	408	634	1
PCSPG-JMP	K100.2	no-pre	21600.7	46672.9	42985	8.58%	439	678	1			
PCSPG-JMP	K100.3	dc	21602.2	28396	28396	0.00%	507	814	3	5		1
PCSPG-JMP	K100.3	no-dc	6.82448	28396	28396	0.00%	507	814	3	493	796	2
PCSPG-JMP	K100.3	no-pre	12.5547	28396	28396	0.00%	507	814	3			
PCSPG-JMP	K100.4	dc	0.658089	29989	29989	0.00%	464	728	2	1	1	0
PCSPG-JMP	K100.4	no-dc	1.05434	29989	29989	0.00%	464	728	2	452	712	1
PCSPG-JMP	K100.4	no-pre	0.642429	29989	29989	0.00%	464	728	2			
PCSPG-JMP	K100.5	dc	0.934127	26339	26339	0.00%	458	716	2	1	2	0
PCSPG-JMP	K100.5	no-dc	15.6518	26339	26339	0.00%	458	716	2	441	698	1
PCSPG-JMP	K100.5	no-pre	0.858563	26339	26339	0.00%	458	716	2			
PCSPG-JMP	K100.6	dc	1.02124	27006	27006	0.00%	407	614	1	4	2	2
PCSPG-JMP	K100.6	no-dc	2.1586	27006	27006	0.00%	407	614	1	393	594	1
PCSPG-JMP	K100.6	no-pre	1.30602	27006	27006	0.00%	407	614	1			
PCSPG-JMP	K100.7	dc	2.43386	39431	39431	0.00%	415	630	1	8	3	1
PCSPG-JMP	K100.7	no-dc	8.29406	39431	39431	0.00%	415	630	1	401	610	1
PCSPG-JMP	K100.7	no-pre	20.4704	39431	39431	0.00%	415	630	1			
PCSPG-JMP	K100.8	dc	4.05139	57765	57765	0.00%	443	686	3	2	1	0
PCSPG-JMP	K100.8	no-dc	7.31388	57765	57765	0.00%	443	686	3	415	650	1
PCSPG-JMP	K100.8	no-pre	35.9737	57765	57765	0.00%	443	686	3			
PCSPG-JMP	K100.9	dc	4.10501	23535	23535	0.00%	433	666	1	1	1	1
PCSPG-JMP	K100.9	no-dc	5.28624	23535	23535	0.00%	433	666	1	415	644	1
PCSPG-JMP	K100.9	no-pre	4.83136	23535	23535	0.00%	433	666	1			
PCSPG-JMP	K200	dc	1073.62	55947	55947	0.00%	891	1382	3	5	4	3
PCSPG-JMP	K200	no-dc	13999.9	70383	39772	76.97%	891	1382	3	867	1354	1
PCSPG-JMP	K200	no-pre	21605.9	75198.2	39772	89.07%	891	1382	3			
PCSPG-JMP	K400	dc	21599.9	109955	48709	125.74%	1915	3030	1	4	3	2
PCSPG-JMP	K400	no-dc	21613.9	100128	48709	105.56%	1915	3030	1	1891	3004	1
PCSPG-JMP	K400	no-pre	21599.9	99824.5	48709	104.94%	1915	3030	1			
PCSPG-JMP	K400.1	dc	21609.8	119541	31868	275.11%	1870	2940	3	7	6	1
PCSPG-JMP	K400.1	no-dc	21609.7	117450	31868	268.55%	1870	2940	3	1858	2928	3
PCSPG-JMP	K400.1	no-pre	21601.1	113363	31479	260.12%	1870	2940	3			
PCSPG-JMP	K400.10	dc	21601	150351	52892	184.26%	1907	3014	4	3	6	11
PCSPG-JMP	K400.10	no-dc	17839.8	148959	50707	193.76%	1907	3014	4	1884	2992	2
PCSPG-JMP	K400.10	no-pre	21600	150305	49414	204.17%	1907	3014	4			
PCSPG-JMP	K400.2	dc	19620.4	149295	41370	260.88%	1927	3054	1	3	4	6
PCSPG-JMP	K400.2	no-dc	21612.4	169679	41370	310.15%	1927	3054	1	1901	3028	1

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
PCSPG-JMP	K400.2	no-pre	21601.9	163632	39442	314.87%	1927	3054	1			
PCSPG-JMP	K400.3	dc	21602.7	102631	28888	255.27%	1892	2984	2	12	7	1
PCSPG-JMP	K400.3	no-dc	21603.7	105073	28888	263.73%	1892	2984	2	1868	2960	1
PCSPG-JMP	K400.3	no-pre	21599.5	98401.2	28888	240.63%	1892	2984	2			
PCSPG-JMP	K400.4	dc	21611.1	119360	43628	173.59%	1826	2852	1	1	3	7
PCSPG-JMP	K400.4	no-dc	21600.3	121569	43488	179.55%	1826	2852	1	1812	2838	1
PCSPG-JMP	K400.4	no-pre	21604.5	131881	38564	241.98%	1826	2852	1			
PCSPG-JMP	K400.5	dc	21599.8	162402	48256	236.54%	1856	2912	2	4	10	3
PCSPG-JMP	K400.5	no-dc	21608.9	156189	50262	210.75%	1856	2912	2	1838	2894	1
PCSPG-JMP	K400.5	no-pre	21600.8	152539	50964	199.31%	1856	2912	2			
PCSPG-JMP	K400.6	dc	21599.5	93723.9	28635	227.31%	1976	3152	2	2	4	10
PCSPG-JMP	K400.6	no-dc	21599.4	90012	28635	214.34%	1976	3152	2	1940	3116	1
PCSPG-JMP	K400.6	no-pre	21599.5	87694.8	28197	211.01%	1976	3152	2			
PCSPG-JMP	K400.7	dc	21612.6	170198	65528	159.73%	1842	2884	2	1	6	6
PCSPG-JMP	K400.7	no-dc	21613.3	175865	64865	171.12%	1842	2884	2	1819	2862	1
PCSPG-JMP	K400.7	no-pre	21619.4	183006	65195	180.71%	1842	2884	2			
PCSPG-JMP	K400.8	dc	21613.6	86236.6	23459	267.61%	1916	3032	3	3	6	5
PCSPG-JMP	K400.8	no-dc	21599.5	92815	22948	304.46%	1916	3032	3	1901	3018	1
PCSPG-JMP	K400.8	no-pre	21602.1	96198.9	22948	319.20%	1916	3032	3			
PCSPG-JMP	K400.9	dc	21602	89578.2	42228	112.13%	1900	3000	2	1	6	5
PCSPG-JMP	K400.9	no-dc	21599.8	110063	42179	160.94%	1900	3000	2	1881	2982	1
PCSPG-JMP	K400.9	no-pre	21599.5	111450	40430	175.66%	1900	3000	2			
PCSPG-JMP	P100	dc	21599.8	1.32E+06	1.31E+06	0.95%	417	634	1	1	1	0
PCSPG-JMP	P100	no-dc	21600.4	1.32E+06	1.31E+06	0.93%	417	634	1	339	486	1
PCSPG-JMP	P100	no-pre	21604.4	1.35E+06	1.30E+06	4.15%	417	634	1			
PCSPG-JMP	P100.1	dc	21600.2	629629	604843	4.10%	384	568	1	1	2	0
PCSPG-JMP	P100.1	no-dc	21599.9	631268	604843	4.37%	384	568	1	327	460	1
PCSPG-JMP	P100.1	no-pre	21606.5	673090	578336	16.38%	384	568	1			
PCSPG-JMP	P100.2	dc	60.5825	547283	547230	0.01%	397	594	1	1	2	0
PCSPG-JMP	P100.2	no-dc	57.3526	547280	547230	0.01%	397	594	1	326	464	1
PCSPG-JMP	P100.2	no-pre	3237.23	547279	547230	0.01%	397	594	1			
PCSPG-JMP	P100.3	dc	5060.22	675391	675323	0.01%	416	632	1	1		0
PCSPG-JMP	P100.3	no-dc	15898.9	675390	675323	0.01%	416	632	1	358	518	1
PCSPG-JMP	P100.3	no-pre	21606.5	709870	673328	5.43%	416	632	1			
PCSPG-JMP	P100.4	dc	4968.29	816180	816099	0.01%	384	568	1	1	1	0
PCSPG-JMP	P100.4	no-dc	21599.7	817662	816099	0.19%	384	568	1	320	462	1
PCSPG-JMP	P100.4	no-pre	21602.8	827203	816099	1.36%	384	568	1			
PCSPG-JMP	P200	dc	21599.5	1.12E+06	1.02E+06	9.74%	787	1174	1	1	3	0
PCSPG-JMP	P200	no-dc	21600.4	1.12E+06	1.01E+06	10.51%	787	1174	1	651	940	1
PCSPG-JMP	P200	no-pre	21617.2	1.14E+06	983639	16.24%	787	1174	1			
PCSPG-JMP	P400	dc	21599.8	2.16E+06	1.67E+06	29.20%	1600	2400	3	1	5	1
PCSPG-JMP	P400	no-dc	21599.5	2.17E+06	1.74E+06	24.74%	1600	2400	3	1559	2360	2
PCSPG-JMP	P400	no-pre	21641.2	2.19E+06	1.52E+06	43.97%	1600	2400	3			
PCSPG-JMP	P400.1	dc	21599.9	3.59E+06	3.01E+06	19.36%	1612	2424	1	1	4	0
PCSPG-JMP	P400.1	no-dc	21599.7	3.65E+06	2.97E+06	23.14%	1612	2424	1	1574	2386	1
PCSPG-JMP	P400.1	no-pre	21696.5	3.69E+06	2.84E+06	30.03%	1612	2424	1			
PCSPG-JMP	P400.2	dc	21599.8	2.96E+06	2.48E+06	19.22%	1596	2392	1	1	7	0
PCSPG-JMP	P400.2	no-dc	21599.9	2.95E+06	2.50E+06	18.11%	1596	2392	1	1548	2344	1
PCSPG-JMP	P400.2	no-pre	21630.8	2.97E+06	2.41E+06	23.42%	1596	2392	1			
PCSPG-JMP	P400.3	dc	21599.7	2.48E+06	2.03E+06	22.10%	1575	2350	2	1	7	0
PCSPG-JMP	P400.3	no-dc	21599.4	2.49E+06	2.04E+06	22.03%	1575	2350	2	1526	2302	1
PCSPG-JMP	P400.3	no-pre	21607.2	2.48E+06	1.97E+06	25.88%	1575	2350	2			
PCSPG-PUCNU	bipe2u	dc	21599.5	26	15	73.33%	5563	10026	1	1		0
PCSPG-PUCNU	bipe2u	no-dc	21599.7	26	16	62.50%	5563	10026	1	5563	10026	1
PCSPG-PUCNU	bipe2u	no-pre	21668.8	26	13	100.00%	5563	10026	1			
PCSPG-PUCNU	cc10-2u	dc	21602.9	67.4444	20	237.22%	6144	10240	1	1		0

set	instance	method	time	UB	LB	gap	nodes	edges	comp	nodes-pre (blocks)	edges-pre (tri+)	comp-pre (tri-)
PCSPG-PUCNU	cc10-2u	no-dc	21600.1	64.425	23	180.11%	6144	10240	1	6144	10240	1
PCSPG-PUCNU	cc10-2u	no-pre	21614	70	18	288.89%	6144	10240	1			
PCSPG-PUCNU	cc3-10u	dc	21603.2	19	5	280.00%	14500	27000	1	1		0
PCSPG-PUCNU	cc3-10u	no-dc	21600.2	19	5	280.00%	14500	27000	1	14500	27000	1
PCSPG-PUCNU	cc3-10u	no-pre	21603.4	19	5	280.00%	14500	27000	1			
PCSPG-PUCNU	cc3-11u	dc	21600.6	35	9	288.89%	21296	39930	1	1		0
PCSPG-PUCNU	cc3-11u	no-dc	21600.3	35	10	250.00%	21296	39930	1	21296	39930	1
PCSPG-PUCNU	cc3-11u	no-pre	21600.9	35	7	400.00%	21296	39930	1			
PCSPG-PUCNU	cc3-12u	dc	21600.8	41	12	241.67%	30240	57024	1	1		0
PCSPG-PUCNU	cc3-12u	no-dc	21600.6	41	12	241.67%	30240	57024	1	30240	57024	1
PCSPG-PUCNU	cc3-12u	no-pre	21608.1	41	11	272.73%	30240	57024	1			
PCSPG-PUCNU	cc3-4u	dc	20.8424	3	3	0.00%	352	576	1	1		0
PCSPG-PUCNU	cc3-4u	no-dc	39.8234	3	3	0.00%	352	576	1	352	576	1
PCSPG-PUCNU	cc3-4u	no-pre	21601.8	5	3	66.67%	352	576	1			
PCSPG-PUCNU	cc3-5u	dc	21600.4	6.66667	4	66.67%	875	1500	1	1		0
PCSPG-PUCNU	cc3-5u	no-dc	5411.22	7	4	75.00%	875	1500	1	875	1500	1
PCSPG-PUCNU	cc3-5u	no-pre	21605.3	8.02219	4	100.55%	875	1500	1			
PCSPG-PUCNU	cc5-3u	dc	11480.9	8.88667	4	122.17%	1458	2430	1	1		0
PCSPG-PUCNU	cc5-3u	no-dc	7145.26	8.875	4	121.88%	1458	2430	1	1458	2430	1
PCSPG-PUCNU	cc5-3u	no-pre	21605.8	9.5	4	137.50%	1458	2430	1			
PCSPG-PUCNU	cc6-2u	dc	21.1733	2	2	0.00%	256	384	1	1		0
PCSPG-PUCNU	cc6-2u	no-dc	42.5771	2	2	0.00%	256	384	1	256	384	1
PCSPG-PUCNU	cc6-2u	no-pre	21604.2	4	2	100.00%	256	384	1			
PCSPG-PUCNU	cc6-3u	dc	21613.1	39.1667	17	130.39%	5097	8736	1	1		0
PCSPG-PUCNU	cc6-3u	no-dc	21610.7	39.0833	17	129.90%	5097	8736	1	5097	8736	1
PCSPG-PUCNU	cc6-3u	no-pre	21603.6	39.875	15	165.83%	5097	8736	1			
PCSPG-PUCNU	cc7-3u	dc	21608.1	119.35	43	177.56%	17495	30616	1	1		0
PCSPG-PUCNU	cc7-3u	no-dc	21600.3	125.917	42	199.80%	17495	30616	1	17495	30616	1
PCSPG-PUCNU	cc7-3u	no-pre	21605.5	120	43	179.07%	17495	30616	1			
PCSPG-PUCNU	cc9-2u	dc	19924.9	21.6292	7	208.99%	2816	4608	1	1		0
PCSPG-PUCNU	cc9-2u	no-dc	21600.1	21.9583	8	174.48%	2816	4608	1	2816	4608	1
PCSPG-PUCNU	cc9-2u	no-pre	21609.5	22.926	5	358.52%	2816	4608	1			
Random-graphs	n200-l1.2-rand-graph	dc	21599.7	29.4119	2.63705	1015.33%	1836	3272	1	1		0
Random-graphs	n200-l1.2-rand-graph	no-dc	21599.9	29.4555	2.64785	1012.43%	1836	3272	1	1836	3272	1
Random-graphs	n200-l1.2-rand-graph	no-pre	21653.9	30.3155	1.97287	1436.62%	1836	3272	1			
Random-graphs	n200-l1.5-rand-graph	dc	21599.8	57.2273	13.273	331.16%	1775	3150	1	1		0
Random-graphs	n200-l1.5-rand-graph	no-dc	21600.7	57.2332	13.1947	333.76%	1775	3150	1	1775	3150	1
Random-graphs	n200-l1.5-rand-graph	no-pre	21662.4	58.135	11.6427	399.33%	1775	3150	1			
Random-graphs	n200-l2.0-rand-graph	dc	21599.7	105.13	52.8956	98.75%	1805	3210	1	1		0
Random-graphs	n200-l2.0-rand-graph	no-dc	21599.4	105.093	50.7885	106.92%	1805	3210	1	1805	3210	1
Random-graphs	n200-l2.0-rand-graph	no-pre	21628.5	105.247	51.2217	105.47%	1805	3210	1			
Random-graphs	n200-l3.0-rand-graph	dc	21600	215.052	131.452	63.60%	1816	3232	1	1		0
Random-graphs	n200-l3.0-rand-graph	no-dc	21601	213.962	132.089	61.98%	1816	3232	1	1816	3232	1
Random-graphs	n200-l3.0-rand-graph	no-pre	21646.8	215.355	131.452	63.83%	1816	3232	1			
Random-graphs	n400-l2.0-rand-graph	dc	21601.3	195.222	69.1851	182.17%	3692	6584	1	1		0
Random-graphs	n400-l2.0-rand-graph	no-dc	21601.9	195.276	70.4878	177.04%	3692	6584	1	3692	6584	1
Random-graphs	n400-l2.0-rand-graph	no-pre	21612.2	196.595	68.9002	185.33%	3692	6584	1			