A Capacity Scaling Heuristic for the Multicommodity Capacitated Network Design Problem

N. Katayama [a,*,1], M. Chen [b], M. Kubo [b]

[a] *Department of Distributions and Logistics Systems, Ryutsu Keizai University, 120 Hirahata, Ryugasaki, Ibaraki, 301-8555,Japan*

[b] *Department of Marine Technology, Tokyo University of Marine Science and Technology, 2-1-6 Etchujima,Koto-ku,Tokyo,135-8533,Japan*

Abstract

In this paper, we propose a capacity scaling heuristic using a column generation and row generation technique to address the multicommodity capacitated network design problem. The capacity scaling heuristic is an approximate iterative solution method for capacitated network problems based on changing arc capacities, which depend on flow volumes on the arcs. By combining a column and row generation technique and a strong formulation including forcing constraints, this heuristic derives high quality results, and computational effort can be reduced considerably. The capacity scaling heuristic offers one of the best current results among approximate solution algorithms designed to address the multicommodity capacitated network design problem.

Key words: network design, multicommodity network,capacitated problem, optimization

1 INTRODUCTION

The multicommodity capacitated network design problem ($MCND$) represents a generic network model for applications in designing the construction

* Corresponding author.
Email address: `katayama@rku.ac.jp`, *Telephone number:*`+81-297-64-0001`, *Fax number:*`+81-297-64-0011` (N. Katayama).

1 Supported by the Japanese Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Scientific Research (C), 16510119, 2006.

and improvement of telecommunication, logistics, transportation, distribution and production networks. For network design problems, a wide range of application models can be found in Magnanti and Wong [1]. The solution of $MCND$ provides the appropriate network design and routes of multicommodity flows to minimize the total cost that is the sum of flow costs and design costs over the network with limited arc capacities. $MCND$ is formulated as a mixed integer programming problem. Binary variables are used to model the network design selecting arcs from a candidate arc set appropriately, while continuous variables represent the volumes of path flows on the network. $MCND$ is known as an NP-hard problem. Therefore, many techniques such as valid inequalities, relaxation methods and heuristics, have been developed.

The polyhedral approach to improve the formulation by adding valid inequalities has been developed. Magnanti, Mirchandan and Vachani [2] proposed integer rounding cut-set, three-partition and arc residual capacity inequalities, and Barahona [3] proposed multi-cut inequalities. Recently Chouman, Crainic and Gendron [4] proposed cover inequalities, minimum cardinality inequalities and these lifting inequalities, as well as cut-set inequalities.

Relaxation and lower bound approaches have been devised for solving $MCND$. Katayama and Kasugai [5] proposed a dual ascent method for integer rounding cut-set inequalities. Gendron and Crainic [6][7] presented linear relaxation and Lagrangian relaxation problems, and proposed solution algorithms. Crainic, Frangioni and Gendron [8] developed a subgradient method using a bundle type algorithm. Holmberg and Yuan [9] proposed a combination algorithm of a Lagrangian relaxation method and a branch-and-bound algorithm. Herrmann et al. [10] proposed an extension method of a dual ascent algorithm for an uncapacitated network design problem.

Heuristics and meta-heuristics designed to find feasible approximate solutions within a reasonable computation time have been developed. Gendron and Crainic [6][7] proposed a resource decomposition heuristic based on a resource-directive decomposition algorithm for a multicommodity network flow problem. Crainic, Gendreau and Farvolden [11] and Zaleta and Socarrás [12] proposed simplex-based tabu search methods. Ghamlouche, Crainic and Gendreau [13] proposed a cycle-based tabu search method combining the simplex-based search. Ghamlouche, Crainic and Gendreau [14] and Álvarez, González and De-Alba [15] proposed path relinking algorithms or scatter search algorithms. Crainic and Gendreau [16] proposed a cooperative parallel tabu search and Crainic, Li and Toulouse [17] proposed a multilevel cooperative search. Recently Crainic, Gendron and Hernu [18] proposed slope scaling heuristics. The slope scale heuristic is based on changing flow costs, which depend on arc flow volumes and dual variable information, and solving multicommodity network flow problems.

This paper presents a capacity scaling heuristic using a path-based formulation including tight forcing constrains and a column generation and row generation technique. In many papers, an arc-flow based formulation is used for $MCND$. Since the arc-flow based formulation including tight forcing constraints is a large mixed integer programming problem, it take significant amounts of time to solve large problem or their linear relaxation problems. Consequently, we use a path-based formulation including tight forcing constraints and a column generation and row generation technique, and we are thereby able to efficiently solve $MCND$ by capacity scaling heuristic.

2 MATHEMATICAL FORMULATION

$MCND$ can be described as follows. $G = (N, A)$ denots a directed network with the set of nodes N and the set of directed arcs A. Let K be the set of commodities using this network. For each commodity $k \in K$, let P^k be the set of paths of commodity k, and d^k the required amount of flow of commodity k from its single origin node to its single destination node.

The following measures characterize arc $(i, j) \in A$: f_{ij} the design cost of including arc (i, j) in the network design; c^k_{ij} the unit variable flow cost for commodity k flowing on arc (i, j), and C_{ij} the limited arc capacity, which must be shared by all the commodities flowing on the arc. The formulation of $MCND$ has two type variables. The first type is a binary design variable, which is defined as $y_{ij} = 1$, if arc (i, j) is included in the network design, $y_{ij} = 0$ otherwise. The second type is a continuous flow variable, which is defined by x^k_p representing the amount of the path flow of commodity k flowing on the path $p \in P^k$. Let δ^p_{ij} be the constant, $\delta^p_{ij} = 1$ if arc (i, j) is included in path p, $\delta^p_{ij} = 0$ otherwise.

The path-based formulation of $MCND$ can be formulated as follows:

$$\textit{minimize} \quad \sum_{(i,j)\in A} \sum_{k\in K} c^k_{ij} \sum_{p\in P^k} \delta^p_{ij} x^k_p + \sum_{(i,j)\in A} f_{ij} y_{ij} \tag{1}$$

$$\textit{subject to} \quad \sum_{p\in P^k} x^k_p = d^k \quad \forall k \in K \tag{2}$$

$$\sum_{k\in K} \sum_{p\in P^k} \delta^p_{ij} x^k_p \le C_{ij} y_{ij} \quad \forall (i, j) \in A \tag{3}$$

$$x^k_p \ge 0 \quad \forall p \in P^k, k \in K \tag{4}$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \tag{5}$$

The objective function (1) is the total cost, the sum of variable flow costs of commodities plus the sum of design costs in a given network design, and should be minimized. Constraints (2) are the flow conservation equations, representing the fact that the sum of path flows of commodity k is equal to the required amount. Constraints (3) provide the capacity constraints, which prohibit flowing if the arc is closed ($y_{ij} = 0$), and allowing for flow up to the arc capacity if the arc is opened ($y_{ij} = 1$). Constraints (4) ensure the non-negativity of continuous variables and constraints (5) force binary variables to assume binary values.

When relaxing binary conditions (5), this linear relaxation problem is reduced to the following shortest path problem with arc length $c_{ij}^k + f_{ij}/C_{ij}, (i,j) \in A, k \in K$, since $\sum_{k \in K} \sum_{p \in P^k} \delta_{ij}^p x_p^k = C_{ij} y_{ij}$ is set at the optimal solution.

$$\textit{minimize} \quad \sum_{k \in K} \sum_{(i,j) \in A} (c_{ij}^k + f_{ij}/C_{ij}) \sum_{p \in P^k} \delta_{ij}^p x_p^k \tag{6}$$

$$\textit{subject to} \quad \sum_{p \in P^k} x_p^k = d^k \quad \forall k \in K \tag{7}$$

$$x_p^k \geq 0 \quad \forall p \in P^k, k \in K \tag{8}$$

This shortest problem is disjoint for each commodity and can be solved separately. But the lower bound derived from this relaxation is very weak, and the gap between the lower bound derived and the upper bound is relatively large.

Constraints (3) can be disaggregated for each commodity.

$$\sum_{p \in P^k} \delta_{ij}^p x_p^k \leq d^k y_{ij} \quad \forall (i,j) \in A, k \in K \tag{9}$$

Constraints (9) are the forcing constraints, which prohibit flowing of commodity k if the arc is closed, and allow for flow up to the required amount if the arc is opened. These forcing constraints are redundant by constraints (3), and the number of constraints is very large. Since these constraints are very tight at the linear relaxation problem, they are added to the formulation in order to improve on the lower bound derived from the linear relaxation problem.

3 CAPACITY SCALING HEURISTIC

The capacity scaling heuristic is an approximate iterative solution method for capacitated network problems based on changing arc capacities, which depend on flow volumes on arcs [19]. When solving the linear relaxation problem of

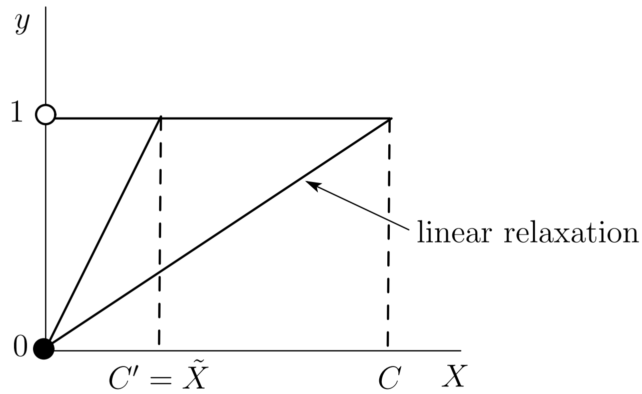

Fig. 1. X-y Relations in a linear relaxation problem

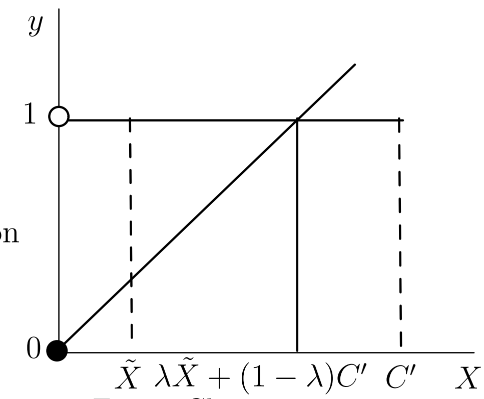

Fig. 2. Changing capacity

$MCND$, the capacity constraints (3), which define a variable upper bound on design variable y for arc flow X, is approximated by a linear function (Figure 1). The design variable is underestimated all over the domain. As a consequence, a relaxation solution may not be good approximation to find a feasible solution of $MCND$. If the optimal flow $\tilde{X}$ of $MCND$ is found, we should change capacity C to C', which is equal to $\tilde{X}$ for each arc. $\tilde{X}$ can be calculated by the equation $\sum_{p \in P^k} \delta_{ij}^p x_p^k$. Then we solve the linear relaxation problem with capacity $\boldsymbol{C'}$ again. As a result, 0 or 1 solutions for all design variables can be obtained. The multicommodity flow problem of all fixed design variables to these solutions is solved, and then the optimal value of $MCND$ may be obtained. As a matter of course, finding the optimal flow of $MCND$ is extremely difficult. If the near optimal flow can be found, $\boldsymbol{C'}$ can be estimated and a good approximate solution might be derived from it. On the other hand, by changing capacity $\boldsymbol{C'}$ a little bit at a time, we seek the near optimal flow.

The capacity scaling heuristic begins by solving the linear relaxation problem of $MCND$ with $\boldsymbol{C'}$ instead of $\boldsymbol{C}$. We set $\boldsymbol{C'}(1) := \boldsymbol{C}$ initially. The linear relaxation problem $LR(\boldsymbol{C'}(l))$ with capacity $\boldsymbol{C'}(l)$ at the iteration l can be formulated as follows:

$$\textit{minimize} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k \sum_{p \in P^k} \delta_{ij}^p x_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \tag{10}$$

$$\textit{subject to} \quad \sum_{p \in P^k} x_p^k = d^k \quad \forall k \in K \tag{11}$$

$$\sum_{k \in K} \sum_{p \in P^k} \delta_{ij}^p x_p^k \leq C_{ij}^{'}(l) y_{ij} \quad \forall (i,j) \in A \tag{12}$$

$$\sum_{p \in P^k} \delta_{ij}^p x_p^k \leq d^k y_{ij} \quad \forall (i,j) \in A, k \in K \tag{13}$$

$$x_p^k \geq 0 \quad \forall p \in P^k, k \in K \tag{14}$$

$$0 \leq y_{ij} \leq C_{ij}/C'_{ij}(l) \quad \forall (i,j) \in A \tag{15}$$

The right-hand side of constraints (12) is changed to $C'_{ij}(l)$, and the right-hand side of constraints (15) is changed to $C_{ij}/C'_{ij}(l)$ to enable flow up to its original capacity C_{ij}.

Let $\tilde{\boldsymbol{X}}$ be the optimal arc flow of $LR(\boldsymbol{C}'(l))$. At the next iteration, we substitute $\boldsymbol{C}'$ by $\lambda\tilde{\boldsymbol{X}} + (1-\lambda)\boldsymbol{C}'$ (Figure 2), where $\lambda(0 \leq \lambda \leq 1)$ is a smoothing parameter preventing rapid jumping. If all design variables converge to zero or one in the solution of $LR(\boldsymbol{C}'(l))$ at some iteration, then we solve the multicommodity network flow problem of all fixed design variables to these values, and a feasible solution to $MCND$ is found. For obtaining converged solutions, it may require numerous iterations, or sometimes they may not be convergent. Consequently when most design variables converge to zero or one by a threshold value ϵ and the number of free design variables is less than a certain number B, then a branch-and-bound algorithm is applied for free variables and the upper bound $Z(l)$ is found. The capacity scaling heuristic stops when the iteration number exceeds the maximum iteration number $MAXN$ and we have found the upper bound UB.

An outline of the capacity scaling heuristic proceeds as follows:

Capacity Scaling Heuristic

1) Set $\lambda \in (0,1)$, ϵ, $MAXN$ and B. $C'_{ij}(1) := C_{ij}, (i,j) \in A$, $UB := \infty$, $l := 1$.
2) Solve $LR(\boldsymbol{C}'(l))$. Let $\tilde{\boldsymbol{X}}$ be the corresponding arc flow and $\tilde{\boldsymbol{y}}$ the corresponding design solution.
3) For each $(i,j) \in A$,

$$\bar{y}_{ij} := \begin{cases} 0 & if \;\; \tilde{y}_{ij} < \epsilon \\ 1 & if \;\; \tilde{y}_{ij} > 1-\epsilon \\ free & otherwise. \end{cases} \tag{16}$$

When the number of free variables of $\bar{\boldsymbol{y}}$ is less than B,
a) Solve the problem with design variable $\bar{\boldsymbol{y}}$ by a branch-and-bound algorithm. Let $Z(l)$ be the corresponding upper bound.
b) If $Z(l) < UB$, then $UB := Z(l)$.
4) If $l \geq MAXN$ and $UB \neq \infty$, then stop the procedure.
5) $l := l+1$. For each $(i,j) \in A$, $C'_{ij}(l) := \lambda\tilde{X}_{ij} + (1-\lambda)C'_{ij}(l-1)$ and change the upper bound of y_{ij} to $C_{ij}/C'_{ij}(l)$. Go to step 2.

4 COLUMN AND ROW GENERATION TECHNIQUE

In the capacity scaling heuristic, the linear programming problem $LR(\boldsymbol{C}'(l))$ is solved iteratively. Since $LR(\boldsymbol{C}'(l))$ has exponentially the large number of path flow variables and has the forcing constraints of the number of $O(|K||A|)$, not every variables and constraint can be included in the model when solving large instances. In order to solve larger instances efficiently, a column generation technique for path flow variables [20] is developed. This technique can also reduce the number of forcing constraints, which can be generated, as needed, via a column generation.

For each commodity k, let $\bar{P}^k \subset P^k$ be the initial set of paths and Δ_{ij}^p the constant, $\Delta_{ij}^p = 1$ if path flow variable $x_p^k, p \in \bar{P}^k$ exists in the formulation and (i,j) is included in path p, $\Delta_{ij}^p = 0$ otherwise. Consequently the only forcing constrains in the formulation is $\sum_{p \in P^k} \Delta_{ij}^p > 0$.

We reformulate the restricted problem with capacity $\boldsymbol{C}'(l)$, restricted path sets $\bar{P}^k, k \in K$ and restricted forcing constraints from $LR(\boldsymbol{C}'(l))$, as following $RLR(\boldsymbol{C}'(l), \bar{P})$;

$$\textit{minimize} \quad \sum_{(i,j)\in A} \sum_{k\in K} c_{ij}^k \sum_{p\in \bar{P}^k} \delta_{ij}^p x_p^k + \sum_{(i,j)\in A} f_{ij} y_{ij} \tag{17}$$

$$\textit{subject to} \quad \sum_{p\in \bar{P}^k} x_p^k = d^k \quad \forall k \in K \tag{18}$$

$$\sum_{k\in K} \sum_{p\in \bar{P}^k} \delta_{ij}^p x_p^k \le C_{ij}^{'}(l) y_{ij} \quad \forall (i,j) \in A \tag{19}$$

$$\sum_{p\in \bar{P}^k} \delta_{ij}^p x_p^k \le d^k y_{ij} \quad \forall (i,j) \in A, k \in K,\ if \sum_{p\in P^k} \Delta_{ij}^p > 0 \tag{20}$$

$$x_p^k \ge 0 \quad \forall p \in \bar{P}^k, k \in K \tag{21}$$

$$0 \le y_{ij} \le C_{ij}/C_{ij}^{'}(l) \quad \forall (i,j) \in A \tag{22}$$

Let $\boldsymbol{s}$ be the dual variable for constraint (18), $\boldsymbol{u}(\ge \boldsymbol{0})$ for constraint (19), $\boldsymbol{w}(\ge \boldsymbol{0})$ for constraint (20). When solving $RLR(\boldsymbol{C}'(l), \bar{P})$ optimally, a dual solution $(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{w})$ is obtained. The reduced cost of path flow variable x_p^k is

$$\sum_{(i,j)\in A} (c_{ij}^k + u_{ij} + w_{ij}^k)\delta_{ij}^p - s^k. \tag{23}$$

The pricing problem is used for generating new path flow variables. The pricing problem of $RLR(\boldsymbol{C}'(l), \bar{P})$ is disjoint for each commodity k and then can be solved separately. The pricing problem for commodity k is written as follows:

$$z^k = minimize \sum_{p \in P^k} \sum_{(i,j) \in A} (c_{ij}^k + u_{ij} + w_{ij}^k) \delta_{ij}^p x_p^k \tag{24}$$

$$subject\ to \sum_{p \in P^k} x_p^k = d^k \tag{25}$$

$$x_p^k \geq 0 \quad \forall p \in P^k \tag{26}$$

Given that $\boldsymbol{u} \geq \boldsymbol{0}$ and $\boldsymbol{w} \geq \boldsymbol{0}$, this is a shortest path problem with nonnegative arc length $c_{ij}^k + u_{ij} + w_{ij}^k, (i,j) \in A$ and can be solved efficiently using Dijkstra's algorithm. Let p^* be the optimal path of this problem. If $z^k < s^k$, then the path flow variable $x_{p^*}^k$ corresponding to the optimal path p^* has negative reduced cost. Then path p^* is added to $\bar{P}^k$, the new variable $x_{p^*}^k$ is generated as a new column, and we have $\Delta_{ij}^{p^*} := 1, (i,j) \in p^*$. Before adding the new path p^*, if $\sum_{p \in P^k} \Delta_{ij}^p = 0, (i,j) \in p^*$ and the forcing constraints do not exist, then they are also generated and added to $RLR(\boldsymbol{C}'(l), \bar{P})$ as new rows.

To summarize, the algorithm with the column and row generation technique solving $LR(\boldsymbol{C}'(l))$ is as follows:

Column and Row Generation Technique

1) For each $k \in K$, set $\bar{P}^k$ and $\Delta_{ij}^p := 1, (i,j) \in A, p \in \bar{P}^k$.
2) Solve $RLR(\boldsymbol{C}'(l), \bar{P})$. Let $(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{w})$ be a corresponding dual solution.
3) For each $k \in K$,
  a) Solve the shortest path problem with the arc length $c_{ij}^k + u_{ij} + w_{ij}^k, (i,j) \in A$. Let z^k be the length of shortest path p^*.
  b) If $z^k < s^k$, then path p^* is added to $\bar{P}^k$, generate path variable $x_{p^*}^k$ and $\Delta_{ij}^{p^*} := 1, (i,j) \in p^*$,
  c) For each $(i,j) \in p^*$, if $\sum_{p \in P^k} \Delta_{ij}^p$ is grater than 0 from 0 in step b), then corresponding forcing constraints are generated and added to $RLR(\boldsymbol{C}'(l), \bar{P})$.
4) If a new path is generated, then go to step 2, otherwise stop the procedure.

5 COMPUTATIONAL EXPERIMENTS

To evaluate the performance of the capacity scaling heuristic proposed in this paper, we compare its output to the optimal value or a lower bound using a branch-and-bound algorithm, as well as to the result of the simplex-based tabu search [11][13], the cycle-based tabu search [13], the path relinking [14] and the multilevel cooperative search [17]. The same two data sets by Crainic

Table 1
Computational results: C problems

METHOD	GAP(%)	METHOD	GAP(%)
SIMPLEX	11.63	SCALE(0.075)	2.52
CYCLE	5.47	SCALE(0.100)	2.54
RELINK	5.08	SCALE(0.125)	2.45
MULTI	4.39	SCALE(0.150)	2.60
SCALE(0.025)	2.55	SCALE(best)	2.28
SCALE(0.050)	2.60		

et al.[11] are used. A detailed description of these problem instances is given in [7][11].

The first set of instances, denoted C, consists of 43 problem instances characterized by the number of nodes, the number of arcs and the number of commodities. Two letters are used to characterize the design cost level, "F" for high and "V" for low relatively to the flow cost, and the capacity level "T" for tight and "L" for loose compared to the total demand. The second set of instances, denoted R, consists of 153 problem instances characterized by three capacity levels, "C1", "C2", "C8", and three design cost levels, "F01", "F05", "F10". If the C value is small, the arc capacities are loose and if large, the arc capacities are tight compared to the total demand. If the F value is small, the design costs are low and if large, the design costs are high compared to the flow costs.

Our experiments were performed on an IBM compatible PC with Pentium 3.2GHz CPU, 1GBytes RAM. The computer code is written in Visual Basic.NET on WINDOWS XP. CPLEX 9.0, a mathematical programming solver by ILOG, is used to solve linear programming problems and mixed integer programming problems in the capacity scaling heuristic. In order to assess the solution quality relative to the optimal values or lower bounds, we solved all instances using the branch-and-bound algorithm of CPLEX and a limit of 10 hours of computation time was imposed for each instance. If the problem cannot be solved optimally within the limit computation time, the best lower bound found in the branch-and-bound algorithm is used instead of the optimal value.

A smoothing parameter λ was calibrated and six values, 0.025, 0.050, 0.075, 0.100, 0.125, 0.150 were tested. A branch-and-bound execution parameter, B=75 for each instance.

Table 1 displays the average of results for the first set, the C problems. Column GAP displays the percentages of the average gap relative to the optimal value/lower bound by CPLEX for the upper bound by each heuristic. SIMPLEX is the result by the simplex-based tabu search, CYCLE by the

Table 2
Computational results: C problems

PROB	OPT /LB	SIMP LEX	CYC LE	REL INK	MUL TI	SCA LE	GAP (%)	IMPR OV(%)
20,100,10VL	14712^{O}	14712	14712	14712	14712	14712	0.00	0.00
20,100,10FL	14941^{O}	15889	14941	14941	14941	15037	0.64	0.64
20,100,10FT	49899^{O}	51654	49899	49899	49937	50771	1.75	1.75
25,100,30VT	365272^{O}	365272	365385	365385	365385	365272	0.00	0.00
25,100,30FL	37055^{O}	38804	37583	37654	37607	37471	1.12	-0.12*
25,100,30FT	85530^{O}	86445	86296	86428	86461	85801	0.32	-0.57*
20,230,40VL	423848^{O}	425046	424778	424385	426702	424075	0.05	-0.07*
20,230,40VT	371475^{O}	371816	371893	371811	371475	371906	0.12	0.12
20,230,40FT	643036^{O}	644172	645812	645548	652894	644483	0.23	0.05
20,300,40VL	429398^{O}	429912	429535	429398	429837	429398	0.00	0.00
20,300,40FL	586077^{O}	589190	593322	590427	593544	587800	0.29	-0.24*
20,300,40VT	464509^{O}	464509	464724	464509	466004	464569	0.01	0.01
20,300,40FT	604198^{O}	606364	607100	609990	619203	604198	0.00	-0.36*
20,230,200VL	92598^{L}	122592	98995	100404	98582	94247	1.78	-4.40*
20,230,200FL	133512^{L}	188590	146535	147988	143150	137642	3.09	-3.85*
20,230,200VT	97344^{L}	118057	104752	104689	102030	97968	0.64	-3.98*
20,230,200FT	132432^{L}	182829	147385	147554	141188	136130	2.79	-3.58*
20,300,200VL	73759^{L}	88398	80819	78184	78210	74913	1.56	-4.18*
20,300,200FL	111655^{L}	151317	123347	123484	121951	115784	3.70	-5.06*
20,300,200VT	74991^{O}	82724	79619	78867	77251	75302	0.42	-2.52*
20,300,200FT	104334^{L}	135593	114484	113584	111173	107858	3.38	-2.98*

O:optimal value; L:lower bound; *:best upper bound is found.

cycle-based tabu search, RELINK by the path relinking, MULTI by the multilevel cooperative search and SCALE(λ) by the capacity scaling heuristic with smoothing parameter λ. SCALE(best) is the result of the best values among all parameters.

Tables 2 and 3 display the detailed results for C problems. Column PROB indicates the number of nodes, arcs, commodities, and the design cost level and the capacity level. Column OPT/LB corresponds to the optimal value/lower bound by CPLEX. "O" indicates that the optimal value is found and "L" indicates that the algorithm stopped due to the time limit condition and this value is a lower bound. Column SCALE displays the best results found among all parameters by the capacity scaling heuristic. Column GAP displays the gaps relative to the optimal value/lower bound by CPLEX for the upper bound by the capacity scaling heuristic. Column IMPROV displays the percentage of improvement of the upper bound by the capacity scaling heuristic relative to the current best upper bound. "$*$" indicates that the best upper bound is

Table 3
Computational results: C problems

PROB	OPT /LB	SIMP LEX	CYC LE	REL INK	MUL TI	SCA LE	GAP (%)	IMPR OV(%)
100,400,10 VL	28423^O	28485	28677	28485	28553	28426	0.01	-0.21*
100,400,10 FL	23949^O	24912	23949	24022	24022	24459	2.13	2.13
100,400,10 FT	59470^L	71128	67014	65278	66284	73566	23.70	12.70
100,400,30 VT	384560^L	385185	385508	384926	385282	384883	0.08	-0.01*
100,400,30 FL	47459^L	58773	51552	51325	50456	51956	9.48	2.97
100,400,30 FT	127825^L	149282	145144	141359	145721	144314	12.90	2.09
30 ,520,100VL	53958^L	56426	54958	54904	55754	54088	0.24	-1.49*
30 ,520,100FL	91285^L	104117	99586	102054	99817	94801	3.85	-4.80*
30 ,520,100VT	51825^L	53288	52985	53017	53512	52282	0.88	-1.33*
30 ,520,100FT	94646^L	107894	105523	106130	102477	98839	4.43	-3.55*
30 ,700,100VL	47603^O	48984	48398	48723	48869	47635	0.07	-1.58*
30 ,700,100FL	58772^L	65356	62471	63091	63756	60194	2.42	-3.64*
30 ,700,100VT	45552^L	47083	47025	47209	47457	46169	1.35	-1.82*
30 ,700,100FT	54233^L	58804	57886	56576	56910	55359	2.08	-2.15*
30 ,520,400VL	111992^L	125831	120652	119416	115671	112846	0.76	-2.44*
30 ,520,400FL	146809^L	177409	161098	163112	156601	149446	1.80	-4.57*
30 ,520,400VT	114237^L	125518	121588	120170	120980	114641	0.35	-4.60*
30 ,520,400FT	150009^L	174526	167939	163675	160217	152744	1.82	-4.66*
30 ,700,400VL	96741^L	110000	106777	105116	102631	97972	1.27	-4.54*
30 ,700,400FL	130724^L	165484	148950	145026	143988	135064	3.32	-6.20*
30 ,700,400VT	94118^L	103768	101672	101212	99195	95306	1.26	-3.92*
30 ,700,400FT	127666^L	150919	142778	141013	138266	130148	1.94	-5.87*

O:optimal value; *L*:lower bound; *:best upper bound is found.

found by the capacity scaling heuristic.

In Table 1, when compared to MULTI, which is the best result among four other heuristics, the capacity scaling heuristic improves the gaps ranging from 1.79% to 2.11%. In Tables 2 and 3, for each instance, the capacity scaling heuristic improves the gaps of maximum 6.20%, and finds the best new solutions for 31 out of 43 problems in set C. The superiority of the capacity scaling heuristic appears to be especially greater when the number of commodities is greater than or equal to 100. For these large difficult problems, the minimum improvement is 1.33% and the maximum 6.20% for the current best upper bound.

Tables 4 and 5 display the computation times in CPU seconds for the capacity scaling heuristic and three other heuristics, computational times of which are reported in the papers. OPT/LB and SCALE were performed with the same

Table 4
Computation times: C problems (seconds)

PROB	OPT/LB	SIMPEX	CYCLE	RELINK	SCALE
20,100,10,VL	0.17	5.60	48.9	12.5	1.3
20,100,10,FL	11.11	8.37	53.8	14.1	7.1
20,100,10,FT	2.91	17.10	51.2	24.1	3.7
25,100,30,VT	0.63	16.57	223.7	101.4	3.2
25,100,30,FL	182.50	33.01	215.4	75.2	14.7
25,100,30,FT	46.45	71.84	224.6	97.0	10.0
20,230,40,VL	1.74	71.29	370.3	148.8	3.0
20,230,40,VT	9.01	90.28	435.6	156.9	3.3
20,230,40,FT	30.11	121.79	423.3	172.2	3.8
20,300,40,VL	1.13	71.05	611.5	224.9	3.2
20,300,40,FL	22.97	113.44	581.9	228.3	6.2
20,300,40,VT	12.69	145.33	589.6	247.9	3.8
20,300,40,FT	11.80	123.42	560.4	214.4	4.4
20,230,200,VL	t	504.50	2663.2	2494.9	442.1
20,230,200,FL	t	491.63	2718.3	2878.3	1658.0
20,230,200,VT	t	548.36	2565.7	2210.9	523.5
20,230,200,FT	t	889.69	3120.1	3385.8	1943.8
20,300,200,VL	t	982.21	4086.8	3566.0	347.7
20,300,200,FL	t	1316.75	4367.9	4012.6	1289.9
20,300,200,VT	35575.48	938.29	3807.9	3924.2	428.7
20,300,200,FT	t	1065.88	4657.5	3857.1	1721.7

PC with 3.2GHz CPU. SIMPLEX was performed by a SUN Ultra60/2300 workstation with 296MHz 2CPUs (one CPU use), 2GBytes RAM, and CYCLE and RELINK were performed by a SUN Enterprise 10000 with 400 MHz 64CPUs (one CPU use), 64GBytes RAM. "t" indicates that the branch-and-bound algorithm stopped due to the time limit condition, and X indicates that no feasible solution can be found. Small instances can be solved optimally by CPLEX, but CPLEX can identify no feasible solution for some large instances. Due to the fact that different CPUs were used, these computation times cannot be compared directly. But the computational times by the capacity scaling heuristic are reasonable and generally short compared to other heuristics.

Table 6 displays the distribution of the average gaps relative to the optimal value/upper bound by CPLEX for the upper bound by each heuristic according to the capacity level and the design cost level in set R. Table 7 displays the same information but according to problem dimensions. In both tables, "1)" indicates that the gaps are calculated by the difference between the op-

Table 5
Computation times: C problems (seconds)

PROB	OPT/LB	SIMPEX	CYCLE	RELINK	SCALE
100,400,10 ,VL	2.33	32.66	336.3	89.2	5.8
100,400,10 ,FL	4752.01	33.00	306.8	82.9	93.3
100,400,10 ,FT	t	81.23	626.5	209.9	55.6
100,400,30 ,VT	t	277.50	1975.3	492.8	17.8
100,400,30 ,FL	t	100.16	1300.6	315.0	621.5
100,400,30 ,FT	t	215.71	1870.0	480.9	153.9
30 ,520,100,VL	12725.17	995.64	3356.0	1194.1	26.9
30 ,520,100,FL	t	939.24	4032.4	1460.0	406.5
30 ,520,100,VT	t	1218.52	3481.1	1513.7	37.3
30 ,520,100,FT	t	670.29	3927.4	1522.7	199.7
30 ,700,100,VL	402.39	1265.11	4396.4	1860.6	38.5
30 ,700,100,FL	t	1479.59	4755.0	1837.5	114.5
30 ,700,100,VT	t	2426.02	4560.1	1894.1	46.3
30 ,700,100,FT	t	1735.72	4866.1	1706.1	96.8
30 ,520,400,VL	t	5789.27	36530.8	27477.4	568.0
30 ,520,400,FL	$t(X)$	6406.62	42929.6	36669.3	2610.4
30 ,520,400,VT	$t(X)$	6522.23	28214.0	23089.1	230.0
30 ,520,400,FT	$t(X)$	8415.24	40010.9	52173.2	1673.9
30 ,700,400,VL	$t(X)$	12636.2	24816.8	22314.8	474.8
30 ,700,400,FL	$t(X)$	11367.7	69540.1	75664.9	1782.9
30 ,700,400,VT	$t(X)$	15879.5	34974.9	24288.9	749.1
30 ,700,400,FT	$t(X)$	11660.4	51877.9	44936.4	1487.1

t:time limit X:no feasible solution is found by CPLEX.

timal value/upper bound by CPLEX and the upper bound by each heuristic. A real gap is calculated by the optimal value/lower bound, but we show this gap to be compared to the results of other heuristics. "2)" indicates that the gaps are calculated by the difference between the optimal value/lower bound by CPLEX and the upper bound by the capacity scaling heuristic. Column CAPACITY indicates the capacity level, Column FIXED COST the design cost level. Column BEST displays the number of problems, the best solutions of which are found by the capacity scaling heuristic, out of the number of problems.

In Table 6, the gaps by the capacity scaling heuristic are smaller than other heuristics in all cases. The average gap relative to the optimal value/upper bound by the capacity scaling heuristic is 0.27% and all gaps are less than 0.8%. The average gap relative to the optimal value/lower bound is 1.10%. The minimum gap is 0.06% and the maximum gap is 2.38%. The best upper bounds are found for more than 60% of problems in each category. The capacity scaling

Table 6
Computational results according to fixed cost and capacity level: R problems

CAPA CITY	FIXED COST	SIMPEX (%)[1)]	CYCLE (%)[1)]	RELINK (%)[1)]	SCALE (%)[1)]	SCALE (%)[2)]	BEST
1	1	2.49	1.48	0.76	0.06	0.06	11/18
1	5	12.31	3.49	2.43	0.29	0.78	12/18
1	10	19.86	3.55	3.09	-0.14	1.46	13/18
2	1	2.21	1.31	0.78	0.09	0.16	11/18
2	5	9.16	3.68	2.64	0.28	0.96	13/18
2	10	14.45	4.27	3.04	0.26	1.80	12/18
8	1	2.96	1.74	1.15	0.33	0.64	9/15
8	5	6.33	4.14	3.23	0.72	1.90	13/15
8	10	8.60	4.40	4.11	0.66	2.38	12/15
AVERAGE/TOTAL		8.87	3.10	-	0.27	1.10	106/153

1):gaps between optimal value/upper bound by CPLEX and upper bound by heuristic

2):gaps between optimal value/lower bound by CPLEX and upper bound by heuristic

heuristic finds the best new solutions for 106 out of 153 problems in set R. In Table 7, the gaps by the capacity scaling heuristic are smaller than other heuristics in most cases, except small instances such as 10 or 25 commodities. The best upper bounds are found for 8 or 9 out of 9 problems in difficult categories such that both the number of commodities and arcs are greater than or equal to 40. All gaps relative to the optimal value/upper bound are less than 1.3%. The minimum gap relative to the optimal value/lower bound is 0.07% and the maximum gap is 3.42%.

The capacity scaling heuristic proposed in this paper performs satisfactorily for the multicommodity capacitated network design problem. By these computational results, we report that the capacity scaling heuristic can offer high quality solutions with a reasonable computation time and improve most current best solutions.

6 CONCLUSION

In this paper, we proposed a capacity scaling heuristic using the column generation and row generation technique for the strong formulation of the multicommodity capacitated network design problem. The performance of the capacity scaling heuristic was evaluated by solving 196 problem instances of two data sets. Computational results are satisfactory and the capacity scaling

Table 7
Computational results according to problem dimensions: R problems

$\|N\|, \|A\|, \|K\|$	SIMPEX $(\%)^{1)}$	CYCLE $(\%)^{1)}$	RELINK $(\%)^{1)}$	SCALE $(\%)^{1)}$	SCALE $(\%)^{2)}$	BEST
10,25 ,10	0.62	0.00	0.00	0.64	0.64	0/6
10,25 ,25	0.57	0.78	0.23	0.11	0.11	3/6
10,25 ,50	0.54	1.63	0.61	0.07	0.07	2/6
10,50 ,10	1.81	0.11	0.08	0.45	0.45	0/9
10,50 ,25	2.78	0.48	0.36	0.52	0.52	2/9
10,50 ,50	7.69	2.47	1.14	0.20	0.20	8/9
10,75 ,10	2.04	0.41	0.04	1.28	1.28	0/9
10,75 ,25	4.03	0.91	0.41	0.79	0.79	3/9
10,75 ,50	8.96	2.87	1.52	0.31	0.31	9/9
20,100,40	5.00	2.78	1.37	0.68	0.68	8/9
20,100,100	9.03	2.69	2.05	0.16	0.19	9/9
20,100,200	8.06	5.12	4.55	0.06	0.49	9/9
20,200,40	8.43	3.51	3.59	1.27	1.60	9/9
20,200,100	16.87	6.33	4.93	-0.01	1.60	9/9
20,200,200	24.41	6.82	5.41	-0.50	1.93	9/9
20,300,40	6.47	2.90	2.08	0.71	1.91	8/9
20,300,100	20.81	5.47	4.68	-0.08	2.73	9/9
20,300,200	23.24	8.20	6.84	-1.85	3.42	9/9

1):gaps between optimal value/upper bound by CPLEX and upper bound by heuristic
2):gaps between optimal value/lower bound by CPLEX and upper bound by heuristic

heuristic finds the best new solutions for 137 out of 196 problem instances.

The capacity scaling heuristic using the linear relaxation problem with forcing constraints can offer high quality results. For combining the column and row generation technique, the computational effort can be reduced considerably. We believe that the capacity scaling heuristic proposed in this paper offers one of the best current results among approximate solution algorithms to resolve the multicommodity capacitated network design problem.

ACKNOWLEGMENTS

The authors wish to thank Professor Teodor Gabriel Crainic, Université du Québec à Montréal, for sending data of multicommodity capacitated network

design problems.

REFERENCES

[1] T. L. Magnanti, P. Mireault, R. T. Wong, Tailoring benders decomposition for uncapacitated network design, Mathematical Programming Study 26 (1986) 112–155.

[2] T. L. Magnanti, P. Mirchandani, R. Vachani, The convex hull of two core capacitated network design problems, Mathematical Programming 60 (1993) 233–250.

[3] F. Barahona, Network design using cut inequalities, SIAM Journal on Computing 6 (1996) 823–837.

[4] M. Chouman, T. G. Crainic, B. Gendron, A cutting-plane algorithm based on cutset inequalities for multicommodity capacitated fixed charge network design, Tech. Rep. CRT-2003-16, Centre de recherche sur les transports,Universite de Montreal (2003).

[5] N. Katayama, H. Kasugai, A capacitated multi-commodity network design problem - a solution method for finding a lower bound using valid inequalities, Journal of Japan Industrial Management Association 44 (1993 in Japanese) 164–175.

[6] B. Gendron, T. Crainic, Relaxations for multicommodity capacitated network design problems, Tech. Rep. CRT-965, Centre de recherche sur les transports,Universite de Montreal (1994).

[7] B. Gendron, T. Crainic, Bounding procedures for multicommodity capacitated fixed charge network design problems, Tech. Rep. CRT-96-06, Centre de recherche sur les transports,Universite de Montreal (1996).

[8] T. Crainic, A. Frangioni, B. Gendron, Bundle-based relaxation methods for multicommodity capacitated fixed charge network design problems, Tech. Rep. CRT-96-45, Centre de recherche sur les transports,Universite de Montreal (1998).

[9] K. Holmberg, D. Yuan, A lagrangian heuristic based branch-and-bound approach for the capacitated network design problem, Operations Research 48 (2000) 461–481.

[10] J. Herrmann, G. Ioannou, I. Minis, J. M. Proth, A dual ascent approach to the fixed-charge capacitated network design problem, European Journal of Operational Research 95 (1996) 476–490.

[11] T. Crainic, M. Gendreau, J. Farvolden, A simplex-based tabu search for capacitated network design, INFORMS Journal on Computing 12 (2000) 223–236.

[12] N. Zaleta, A.M.A.Socarrás, Tabu search-based algorithm for capacitated multicommodity network design problems, 14th International Conference on Electronics, Communications and Computers (2004) 144 – 148.

[13] I. Ghamlouche, T. Crainic, M. Gendreau, Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design, Operations Research 51 (2003) 655–667.

[14] I. Ghamlouche, T. G. Crainic, M. Gendreau, Path relinking, cycle-based neighborhoods and capacitated multicommodity network design, Annals of Operations Research 131 (2004) 109–134.

[15] A. M. Alvarez, J. L. González-Velarde, K. De-Alba, Scatter search for network design problem, Annals of Operations Research 138 (1) (2005) 159–178.

[16] T. G. Crainic, B. Gendron, Cooperative parallel tabu search for capacitated network design, Journal of Heuristics 8 (2002) 601–627.

[17] T.G.Crainic, Y. Li, M. Toulouse, A first multilevel cooperative algorithm for capacitated multicommodity network design, Computers & Operations Research 33 (2006) 2602–2622.

[18] T. Crainic, B. Gendron, G. Hernu, A slope scaling/lagrangean perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design, Tech. Rep. CRT-2003-05, Centre de recherche sur les transports,Universite de Montreal (2003).

[19] Y. Pochet, M. Vyve, A general heuristic for production planning problems, Journal on Computing 16 (2004) 316 – 327.

[20] J. Farvolden, W. Powell, A primal partitioning solution for the arc-chain formulation of a multicommodity network flow problem, Operations Research 41 (1993) 669–693.

[21] I. Ghamlouche, T. Crainic, M. Gendreau, Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design, Tech. Rep. CRT-2001-01, Centre de recherche sur les transports,Universite de Montreal (2001).

A APPENDIX

The detailed results for R problems are given in Tables A.1 to A.5. Column PROB is the type of problems and C/F indicates the capacity level and the design cost level. Column OPT/LB corresponds to the optimal value/lower bound by CPLEX, CYCLE is the result of the cycle-based tabu search[21], and SCALE is the result of the capacity scaling heuristic. Column TIME displays the computation times in CPU seconds for the capacity scaling heuristic and GAP displays the percentages of the average gap relative to the optimal value/lower bound by CPLEX for the upper bound by the capacity scaling heuristic.

Table A.1
Computation results: R problems

PROB	$\|N\|,\|A\|,\|K\|$	C/F	OPT/LB	CYCLE	SCALE	TIME	GAP
R01	10,25,10	1/1	74079^{O}	74079	74079	0.2	0.00%
		1/5	92403^{O}	92403	92403	0.2	0.00%
		1/10	115304^{O}	115304	115304	0.7	0.00%
		2/1	84908^{O}	84908	85146	1.1	0.28%
		2/5	113036^{O}	113036	114565	1.3	1.35%
		2/10	147599^{O}	147599	150821	1.5	2.18%
R02	10,25,25	1/1	232239^{O}	232239	232239	1.2	0.00%
		1/5	322453^{O}	328005	323861	1.4	0.44%
		1/10	419503^{O}	426866	419503	1.8	0.00%
		2/1	316437^{O}	316437	316437	1.2	0.00%
		2/5	431250^{O}	433442	432224	1.4	0.23%
		2/10	559578^{O}	563570	559578	1.4	0.00%
R03	10,25,50	1/1	484830^{O}	484830	484830	1.6	0.00%
		1/5	703362^{O}	712008	704893	2.1	0.22%
		1/10	944990^{O}	981656	944990	2.1	0.00%
		2/1	704247^{O}	706223	704247	0.7	0.00%
		2/5	932897^{O}	953877	932897	2.0	0.00%
		2/10	1188638^{O}	1214120	1190999	1.8	0.20%
R04	10,50,10	1/1	31730^{O}	31730	31730	0.2	0.00%
		1/5	48920^{O}	48920	48920	0.2	0.00%
		1/10	63767^{O}	63767	63767	0.2	0.00%
		2/1	33740^{O}	33740	33760	1.1	0.06%
		2/5	53790^{O}	53790	53790	1.1	0.00%
		2/10	74030^{O}	74030	75109	1.3	1.46%
		8/1	68292^{O}	68293	68512	1.6	0.32%
		8/5	113004^{O}	113226	113226	1.2	0.20%
		8/10	163208^{O}	164430	166453	1.4	1.99%
R05	10,50,25	1/1	123003^{O}	123003	123003	1.1	0.00%
		1/5	170060^{O}	170467	170467	1.6	0.24%
		1/10	221486^{O}	221486	222904	2.8	0.64%
		2/1	131608^{O}	131608	131797	1.2	0.14%
		2/5	204157^{O}	205764	204593	2.2	0.21%
		2/10	286524^{O}	292244	292341	5.3	2.03%
		8/1	278372^{O}	278372	278372	2.1	0.00%
		8/5	445810^{O}	449477	445913	2.0	0.02%
		8/10	625879^{O}	629040	634688	2.3	1.41%

O:optimal value

Table A.2
Computation results: R problems

PROB	$\lvert N\rvert,\lvert A\rvert,\lvert K\rvert$	C/F	OPT/LB	CYCLE	SCALE	TIME	GAP
R06	10,50,50	1/1	245936^O	248615	245936	1.8	0.00%
		1/5	401685^O	412283	401685	3.4	0.00%
		1/10	559477^O	578752	559477	7.6	0.00%
		2/1	286682^O	288460	287580	3.9	0.31%
		2/5	498266^O	515967	501733	10.0	0.70%
		2/10	734414^O	771683	739263	27.6	0.66%
		8/1	682921^O	683614	683039	2.7	0.02%
		8/5	1030479^O	1051780	1030479	0.6	0.00%
		8/10	423316^O	438860	423688	5.7	0.09%
R07	10,75,10	1/1	32807^O	32807	32807	0.2	0.00%
		1/5	47252^O	47252	47252	0.5	0.00%
		1/10	62962^O	62962	62962	1.1	0.00%
		2/1	37432^O	37432	37432	1.2	0.00%
		2/5	56475^O	56591	56915	1.2	0.78%
		2/10	77249^O	78875	79131	1.6	2.44%
		8/1	59947^O	59947	60111	1.9	0.27%
		8/5	99194^O	100155	104161	2.8	5.01%
		8/10	141692^O	142319	145935	1.8	2.99%
R08	10,75,25	1/1	102531^O	102556	102645	1.3	0.11%
		1/5	143894^O	143894	143894	1.6	0.00%
		1/10	182793^O	182793	182793	1.2	0.00%
		2/1	109325^O	109325	109325	1.3	0.00%
		2/5	157047^O	158168	157720	2.3	0.43%
		2/10	207540^O	208135	208467	3.9	0.45%
		8/1	154160^O	155384	156158	3.6	1.30%
		8/5	274867^O	283133	280887	4.1	2.19%
		8/10	415793^O	429896	426900	5.7	2.67%
R09	10,75,50	1/1	171512^O	172343	171923	2.0	0.24%
		1/5	296712^O	307038	298155	5.2	0.49%
		1/10	424266^O	435590	424266	9.3	0.00%
		2/1	192736^O	193242	192833	2.3	0.05%
		2/5	357318^O	371998	357318	9.6	0.00%
		2/10	522187^O	555945	526266	10.9	0.78%
		8/1	345057^O	348297	345646	3.0	0.17%
		8/5	646579^O	669802	647178	4.4	0.09%
		8/10	951136^O	987938	960306	5.2	0.96%

O:optimal value

Table A.3
Computation results: R problems

PROB	$\|N\|,\|A\|,\|K\|$	C/F	OPT/LB	CYCLE	SCALE	TIME	GAP
R10	20,100,40	1/1	200087^{O}	200613	200087	2.4	0.00%
		1/5	346814^{O}	350573	351173	9.9	1.26%
		1/10	488015^{O}	507118	492409	12.0	0.90%
		2/1	229196^{O}	232473	229513	4.0	0.14%
		2/5	411664^{O}	432913	418396	8.0	1.64%
		2/10	609104^{O}	640621	612598	37.4	0.57%
		8/1	486895^{O}	488737	487844	17.9	0.19%
		8/5	951056^{O}	980010	960538	19.1	1.00%
		8/10	1421740^{O}	1487270	1428365	14.8	0.47%
R11	20,100,100	1/1	714431^{O}	725416	714431	10.8	0.00%
		1/5	1263713^{O}	1306090	1268235	152.6	0.36%
		1/10	1843611^{O}	1914040	1854830	327.5	0.61%
		2/1	870451^{O}	876894	871275	20.6	0.09%
		2/5	1623640^{O}	1694860	1625505	80.9	0.11%
		2/10	2414060^{O}	2607690	2427207	237.3	0.54%
		8/1	2294912^{O}	2295790	2295439	9.3	0.02%
		8/5	3507100^{O}	3568430	3507100	10.9	0.00%
		8/10	4579353^{O}	4621900	4579353	8.5	0.00%
R12	20,100,200	1/1	1639443^{O}	1713670	1640889	36.5	0.09%
		1/5	3360268^{L}	3746250	3411185	1635.9	1.52%
		1/10	5144559^{L}	6070200	5283791	842.3	2.71%
		2/1	2303557^{O}	2326230	2305090	50.6	0.07%
		2/5	4669799^{O}	4967940	4669799	37.1	0.00%
		2/10	7100019^{O}	7638050	7100019	33.6	0.00%
		8/1	7635270^{O}	7637250	7635270	28.8	0.00%
		8/5	10067742^{O}	10121700	10067742	8.5	0.00%
		8/10	11967768^{O}	12079300	11967768	5.8	0.00%
R13	20,200,40	1/1	142947^{O}	144138	143036	3.5	0.06%
		1/5	263800^{O}	270316	265049	47.6	0.47%
		1/10	365836^{O}	374999	370229	67.8	1.20%
		2/1	150977^{O}	151513	151170	4.8	0.13%
		2/5	282682^{O}	291510	284213	49.0	0.54%
		2/10	406790^{O}	420028	412045	178.6	1.29%
		8/1	208088^{O}	212451	209579	16.5	0.72%
		8/5	441944^{L}	484112	463356	26.0	4.84%
		8/10	686486^{L}	758715	721677	88.1	5.13%

O:optimal value; L:lower bound

Table A.4
Computation results: R problems

PROB	$\|N\|, \|A\|, \|K\|$	C/F	OPT/LB	CYCLE	SCALE	TIME	GAP
R14	20,200,100	1/1	403414^O	415119	404310	14.1	0.22%
		1/5	749429^L	803356	753409	144.0	0.53%
		1/10	1040970^L	1155840	1067891	346.2	2.59%
		2/1	437607^O	453204	438261	15.4	0.15%
		2/5	839290^L	912456	857209	463.7	2.14%
		2/10	1194757^L	1333440	1216473	660.8	1.82%
		8/1	665247^L	702226	669847	111.0	0.69%
		8/5	1587532^L	1748930	1630867	466.3	2.73%
		8/10	2577261^L	2882710	2669279	171.7	3.57%
R15	20,200,200	1/1	1000787^O	1049360	1000787	40.4	0.00%
		1/5	1912558^L	2158720	1977671	1290.1	3.40%
		1/10	2738164^L	3135760	2908952	5105.0	6.24%
		2/1	1146858^L	1215130	1149298	169.5	0.21%
		2/5	2410765^L	2756680	2484342	2103.5	3.05%
		2/10	3696226^L	4384640	3850096	3409.0	4.16%
		8/1	2297691^L	2355730	2301798	84.6	0.18%
		8/5	5573413^O	5926330	5581719	39.6	0.15%
		8/10	8696932^O	9180920	8696932	65.6	0.00%
R16	20,300,40	1/1	136161^O	136538	136161	1.7	0.00%
		1/5	239500^O	247682	240221	34.2	0.30%
		1/10	325671^O	338807	325839	158.5	0.05%
		2/1	138532^O	139973	138532	6.4	0.00%
		2/5	241801^O	246014	241801	55.3	0.00%
		2/10	337762^O	355610	342618	173.4	1.44%
		8/1	168951^L	172268	173387	9.8	2.63%
		8/5	339516^L	365214	359062	77.5	5.76%
		8/10	509101^L	569874	544884	177.0	7.03%

O:optimal value; L:lower bound

Table A.5
Computation results: R problems

PROB	$\|N\|, \|A\|, \|K\|$	C/F	OPT/LB	CYCLE	SCALE	TIME	GAP
R17	20,300,100	1/1	354138^O	370090	354223	21.7	0.02%
		1/5	643636^L	688554	655289	363.1	1.81%
		1/10	874666^L	971151	920050	1573.8	5.19%
		2/1	370590^O	380850	370622	26.2	0.01%
		2/5	698457^L	753188	712283	324.3	1.98%
		2/10	973365^L	1108180	1032829	1207.4	6.11%
		8/1	497323^L	524038	504634	154.0	1.47%
		8/5	1079856^L	1195140	1111289	1276.5	2.91%
		8/10	1719164^L	1945080	1805653	954.1	5.03%
R18	20,300,200	1/1	828034^L	872888	831435	190.6	0.41%
		1/5	1500315^L	1716680	1546742	2154.2	3.09%
		1/10	2076541^L	2377560	2204809	4203.4	6.18%
		2/1	912710^L	975396	923609	165.6	1.19%
		2/5	1761083^L	2037950	1832785	16556.7	4.07%
		2/10	2577531^L	2966370	2737612	42723.9	6.21%
		8/1	1458573^L	1622520	1483219	2451.1	1.69%
		8/5	3776540^L	4576750	3914013	853.9	3.64%
		8/10	6143618^L	7504310	6409475	287.0	4.33%

O:optimal value; L:lower bound