Results and Discussions

In order to illustrate the application of the two solution approaches, Dinkelbach's algorithm and general MINLP methods, we present computational experiments for 20 instances. The computational experiments are carried out on an Intel 3.2 GHz machine with 512 MB memory. The models are coded in GAMS 22.8.1. The MILP problems in Dinkelbach's solution procedure are solved using CPLEX 11.1. The MINLP solvers include SBB (special branch-and-bound algorithm), DICOPT (outer-approximation algorithm), and α-ECP (α - extended cutting-plane method), and the global optimizer used in the computational experiments is BARON 8.1.4. The relative optimality tolerance is set to be 10^{-6}.

We solve the MILFP problems (CE) as follows:

$$\text{(CE)} \quad \max \quad \frac{a_0 + \sum_{i \in I} a1_i x_i + \sum_{j \in J} a2_j y_j}{b_0 + \sum_{i \in I} b1_i x_i + \sum_{j \in J} b2_j y_j}$$

$$\text{s.t.} \quad c_{0k} + \sum_{i \in I} c1_{ik} x_i + \sum_{j \in J} c2_{jk} y_j = 0, \qquad \forall k \in K$$

$$b_0 + \sum_{i \in I} b1_i x_i + \sum_{j \in J} b2_j y_j \geq 0.001$$

$$0.1 \leq x_i \leq 1, \ \forall i \in I \ \text{ and } \ y_j \in \{0,1\}, \ \forall j \in J$$

which includes $|I|$ continuous variables, $|J|$ binary variables, and $|K|+1$ constraints/equations. The values of $|I|$, $|J|$ and $|K|$ range from 100 to 2,000. The input data are generated randomly as follows: a_0 is generated uniformly on U[-10, 10], b_0 is generated uniformly on U[-10, 10], $a1_i$ is generated uniformly on U[0, 10], $a2_j$ is generated uniformly on U[0, 10], $b1_i$ is generated uniformly on U[0, 10], $b2_j$ is generated uniformly on U[0, 10], c_{0k} is generated uniformly on U[-15, 20], $c1_{ik}$ is generated uniformly on U[-10, 10], $c2_{jk}$ is generated uniformly on U[-20, 15].

For each instance, we solve Problem (CE) with Dinkelbach's algorithm using CPLEX (the initial value of q is set to 0), and the MINLP solvers SBB, DICOPT, α-ECP, as well as the global optimizer BARON. For all the instances the random numbers of input data are generated with the same seed (the default value of GAMS, 3141), so that all the results are reproducible. The computational results are given in Table 1. The optimal solutions obtained with all the methods are the same and equal to the global maximum, although DICOPT does not guarantee global optimality. The

CPU times vary from one approach to the other. Specifically, solving MILFP problems with Dinkelbach's algorithm requires up to 7 iterations and it is usually faster than using the global optimizer BARON and the MINLP solver α-ECP. Dinkelbach's algorithm usually requires slightly longer computational times than SBB and DICOPT for medium instances, but usually less CPU time than these MINLP solvers for large scale instances. More importantly, for very large scale instances with up to thousands of constraints and variables (instances 18-20), DICOPT and SBB both exceed the memory limit while Dinkelbach's algorithm can still return global optimal solutions. This is due to the smaller memory requirement when solving MILP rather than MINLP problems. As for the comparison between the MINLP solvers and the global optimizer, SBB and DICOPT require similar computational times and are both much faster than α-ECP and BARON, especially for large scale instances (instances 11-17). The computational studies suggest that SBB and DICOPT might be the most efficient solvers for medium size MILFP problems, and Dinkelbach's algorithm may have higher computational efficiency for very large scale instances. In all runs, DICOPT was able to find the global optimal solutions, despite the fact that the outer-approximation method cannot guarantee global optimality for pseudoconvex/pseudoconcave functions. The results also show that Dinkelbach's algorithm with CPLEX and the convex MINLP solvers (SBB, DICOPT and α-ECP) are far more efficient than global optimizer BARON for solving MILFP problems.

Table 1 Computational results for MILFP Problems with Dinkelbach's algorithm and MINLP methods

No.	Cont. Var. |*I*|	Disc. Var. |*J*|	Constr. |*K*|+1	Dinkelbach's Algorithm (CPLEX)			SBB		DICOPT		α-ECP		BARON	
				Iter.	Obj.	CPU (s)	Obj.	CPU (s)	Obj.	CPU (s)	Obj.	CPU (s)	Obj.	CPU (s)
1	100	100	101	6	3.033	0.59	3.033	0.11	3.033	0.06	3.033	5.00	3.033	0.69
8	200	100	101	5	2.213	0.65	2.213	0.08	2.213	0.08	2.213	7.00	2.213	10.97
9	300	100	101	6	2.267	1.45	2.267	0.49	2.267	0.87	2.267	161.0	2.267	34.48
10	500	100	101	5	2.085	2.16	2.085	0.53	2.085	1.48	2.085	18.98	2.085	71.64
2	100	200	101	6	3.256	0.97	3.256	0.06	3.256	0.08	3.256	6.00	3.256	11.68
3	100	300	101	7	3.556	1.44	3.556	0.47	3.556	0.87	3.556	65.40	3.556	31.05
4	100	500	101	6	3.691	1.62	3.691	0.39	3.691	0.11	3.691	18.72	3.691	36.40
5	100	100	201	6	3.033	0.84	3.033	0.19	3.033	0.06	3.033	8.00	3.033	12.69
6	100	100	301	6	3.033	1.34	3.033	0.11	3.033	0.08	3.033	12.00	3.033	28.03
7	100	100	501	6	3.033	2.58	3.033	0.14	3.033	0.16	3.033	26.88	3.033	25.72
11	500	500	501	6	2.634	15.79	2.634	5.59	2.634	12.19	2.634	852.70	2.593~3.225*	3,600*
12	1,000	500	501	6	2.311	68.11	2.311	50.31	2.311	45.11	1.786*	3,600*	2.310~14.811*	3,600*
13	500	1,000	501	6	2.887	24.29	2.887	0.92	2.887	0.90	2.887	1,460	2.887~5.872*	3,600*
14	500	500	1,001	6	2.633	34.11	2.633	1.77	2.633	1.77	2.633	1,647	2.633~19.205*	3,600*
15	1,000	1,000	1,001	7	2.608	222.22	2.608	146.57	2.608	164.09	2.430*	3,600*	---	3,600**
16	2,000	1,000	1,001	6	2.310	63.58	2.310	839.28	2.310	468.67	---	3,600**	---	3,600**
17	1,000	2,000	1,001	6	3.085	69.92	3.085	963.50	3.085	888.97	---	3,600**	---	3,600**
18	1,000	1,000	2,001	7	2.603	1466.59	---**	>memo	---**	> memo	---	3,600**	---	3,600**
19	2,000	2,000	1,001	6	2.593	517.95	---**	>memo	---**	> memo	---	3,600**	---	3,600**
20	2,000	2,000	2,001	6	2.705	242.23	---**	>memo	---**	> memo	---	3,600**	---	3,600**

***: Suboptimal solutions (or lower and upper bounds for BARON) obtained after 1 hour (3,600 seconds)**
****: No solution was returned after 1 hour (3,600 seconds)**
>memo: terminates because memory upper limit exceed