Sydney CK Chu

Christina SY Yuen

EFFECTIVE HEURISTICS VS GP SOLUTIONS FOR SHIFT DUTIES GENERATION*

INTRODUCTION

This paper reports comprehensive computation experiences on a general modeling framework for a flexible crew-shift duties generation problem (DGP) initially described in [5]. DGP arises naturally as a mathematical description for the crew (being bus drivers) deployment problem against the background of projects [13] conducted at the Busing and Baggage Departments of the Hongkong Airport Services (HAS), Ltd. HAS of the Hong Kong International Airport is the primary handler of all ground services and aircrafts support functions. Our resulting goal programming (GP) approach has, for the actual case study, exhibited its significant impact on the manpower planning issues albeit its apparent modeling simplicity [9]. The primary factor for its success is GP models' ease of handling frequent changes of flight schedules by modeling flexibility in the work patterns of workers' fixed-length duties [13].

Beyond case studies, there are two natural concerns on the GP models further examined in this paper: Concern on model robustness (or its modeling adaptiveness for different problem scenarios) and concern on computation robustness with *integer* variables. The first concern is addressed here by way of successful computations with different key control parameter values for 25 sets of randomly generated problem instances of input data. The second concern is then further addressed in terms of a "best-fitting" type heuristics and its comparative study (with GP) on the same 25 data sets. Finally the heuristics is tested on an extensive set of 1 000 additional randomized data instances.

Since the main focus of this paper is on computational experience, a detailed review on the vast literatures on manpower duties and crew planning/scheduling/rostering problems (DGP/CSP/CRP) is not given here. Instead, we mention two key review references: a classical review of Bodin, Golden, As-

*This work is partially supported by the HKU Small Project Funding (10205106/06772/25500 /323/01). Virtual-BASIC programming and spreadsheets for the heuristics are carried out by our former research assistant and graduate student, Miss Christina Yuen.

sad & Ball [3]; and a most updated collection of papers given in a whole issue of the ”European Journal of Operational Research” (EJOR) 2004, Vol. 153(1). This February 2004 feature issue of EJOR provides a comprehensive review of the areas of “Timetabling and Rostering”. Significant scientific interests are evidenced by the success of the EURO Working group on Automated Timetabling (WATT) and the international series of conferences on the Practice and Theory of Automated Timetabling (PATAT). A dozen or so papers in this special issue report on a wide range of rostering applications, with an editorial by Burke and Petrovic [4]. Examples include review paper of staff scheduling and rostering by Ernst et al. [10]; nurse rostering problem by Bellanti et al. [2]; local search for shift design by Musliu et al. [11]; and a case study of single shift planning and scheduling by Azmat and Widmer [1].

For an overview of DGP/CSP/CRP as such that is more closely related to our specific airport applications, the readers can refer to our forthcoming article to appear in a future issue of EJOR [7] or its conference proceedings versions of Chu [6] and Chu and Yuen [9].

1. GOAL PROGRAMMING FORMULATION

The modeling formulation of DGP that we describe here can be interpreted as the basic core – the planner – of a more sophisticated DGP/CSP/CRP integrated model in the following sense. DGP in its simplest form (computes and) allocates duties (of given fixed structure of work pattern, rather than crew or staff needing further varying requirements of scheduling) to cover known demands. Demands are given, for equally spaced (hourly) time intervals of (the working time of) a day.

As such, DGP is the *prerequisite* to CSP and CRP in that it provides the planning inputs needed in subsequent scheduling and rostering of staff. As its name implies, DGP allocates duties (performed by crew) in an optimal way to meet known demand over a contiguous number of time intervals. We study its *base* formulation in this paper as stated below. A more detailed account of DGP with its extensions is given in an earlier paper of Chu [5] mentioned above.

We use the following notations for our GP model. Let H be the working time horizon, and let $h = 1, \cdots, H$ index the individual hours. R_h denotes the demand for interval h and d_h represents the over allocation (or over-achievement deviation variable in a GP context) at interval h. The length of a duty is denoted by J. The primary decision variable x_{ij} is the number of allocated staff that starts duty from interval i and breaks at the j^{th} interval after the start of duty, $j = 1, \cdots, J$. Hence for a working horizon of intervals $1, \cdots, H$, we have for the index $i = S, \cdots, T$. The earliest start interval S is such that $S \geqslant 1$ whereas the latest start interval T is limited to $T \leqslant H - J + 1$ (to finish work at interval H). Note that normally $S = 1$ as long as $R_1 > 0$ (there is demand for the very first interval); and $T = H - J + 1$ whenever $R_H > 0$ (there is demand for the very last interval).

We are now ready to state the base model of DGP in terms of a (linear integer goal) programming formulation:

$$\text{Min} \qquad \sum_{i=S}^{T} \sum_{j=1}^{J} c_{ij} x_{ij} \quad + \quad WD \tag{0}$$

$$\text{Subject to} \quad \sum_{i=p}^{q} \sum_{j \neq h-i+1} x_{ij} \quad - \quad d_h \quad = \quad R_h \ , \qquad h = 1, \cdots, H \tag{1}$$

$$d_h \ \leqslant \ D \ , \qquad h = 1, \cdots, H \tag{2}$$

where $p \equiv \max\ \{h - J + 1, S\}$, $q \equiv \min\ \{h, T\}$, and the allocation plan $\{\ x_{ij}\ \}$ are non-negative integer variables.

We see that the LHS of constraint (1) is the total work contribution as a function of $\{\ x_{ij}\ \}$ since the summation over $j = 1, \cdots, J$ $(j \neq h - i + 1)$ spans the $J - 1$ working intervals while the summation over $p \leqslant i \leqslant q$ picks out the total number of staff covering interval h. The single variable D of constraint (2) records the maximum (i.e. over achievement) deviation over all time intervals. Hence (for a "smoothed" allocation) it is minimized, either non-preemptively with a weighting factor W as shown in (0) here, or preemptively as the second priority goal. The coefficients $\{\ c_{ij}\ \}$ can either represent the actual unit pays of staff or (V-shaped) weighting parameters for different time and/or meal break intervals.

Lastly, we give a brief explanation of the summation indices of i ranging from p to q in (1): At time interval h, the index i in x_{ij} would lead to a "covering" duty (i.e. x_{ij} contributing workforce supply at time h), if i satisfies $i + J - 1 \geqslant h$ (from the *earliest* possible start of time i). This implies $i \geqslant h - J + 1$. Similarly, i satisfies $i \leqslant h$ (to the *latest* possible start of time i). Hence for index i to cover time h, we must have $h - J + 1 \ \leqslant \ i \ \leqslant \ h$. Therefore, together with $S \leqslant i \leqslant T$, we have

$$p \equiv \max\ \{h - J + 1, S\} \ \leqslant \ i \ \leqslant \ \min\ \{h, T\} \ \equiv \ q,$$

as shown in (1) above.

As an illustration of its computation, we show, in Figure 1, a typical numerical (daily) output from the DGP computation using a simple Lingo code [12] for our bus drivers' fixed-length duties generation application. For this actual problem instance, the parameters used are: $H = 19, I = 11, J = 9, S = 1, T = 11, c_{ij} = 1$ (implying uniform pay rate) and $W = 1000$. Results for a total of 19 hourly time intervals are shown in Figure 1. In the figure, every 3 occurences of the symbol "#" denote one unit of manpower demand; whereas the counterparts of the symbol "0" refer to one unit of manpower over-allocation, for a specific time interval (each row).

```
=======================================================
Time-interval vs Demand(#) and Over-allocation(0)
-------------------------------------------------------
01 #########                                    ( 3/ 0)
02 ##############################               (10/ 0)
03 ##################000000000000               ( 6/ 4)
04 #################################            (11/ 0)
05 #########000000000000000                     ( 3/ 5)
06 #########000000000000000                     ( 3/ 5)
07 ##################000000000000               ( 6/ 5)
08 ############000000000000000                  ( 4/ 5)
09 #########000000000000000                     ( 3/ 5)
10 ##################000000                     ( 6/ 2)
11 ###############000000000000000               ( 5/ 5)
12 ############################000000000000     ( 8/ 4)
13 ########################################     (12/ 0)
14 ############################000000000000     ( 8/ 4)
15 ###############################000000000     ( 9/ 3)
16 ###############################000000000     ( 9/ 3)
17 ###############000000000000000               ( 5/ 5)
18 #########000000000000000                     ( 3/ 5)
19 ###############000000000000000               ( 5/ 5)
-------------------------------------------------------
Note: Time 01 = 0500-0600hr, .. , Time 19 = 0000-0100hr
```

Fig. 1. Calculated allocation vs demand (output of DGP)

A duty amounts to a continuous stretch of work contribution (from a single staff) for a number of time intervals, with a gap of one meal break. At certain time interval(s), the calculated allocation(s) will be over and above the corresponding demand(s). An optimal DGP solution nevertheless minimizes such total over allocation. This (daily) result in Figure 1 leads to a total over allocation ($\sum d_h$ in (1) above) of 65 man-hours. This yields, with a total demand ($\sum R_h$ in (1) above) of 119, an effective overall 'utilization' (ratio) of

$$\text{Ratio} \equiv 100 \times \frac{\sum R_h}{\sum R_h + \sum d_h} = 64.67\%.$$

We remark that this performance measure of Ratio is directly (or inversely) proportional to the base model's first criterion objective function value $\sum c_{ij} x_{ij}$, where the coefficients c_{ij} are taken to be 1 throughout this paper. This fact can be readily seen as follows. From (0), we have

$$\text{Min} \sum x_{ij} = \text{Max} \frac{1}{\sum x_{ij}} ;$$

and from (1), summing over all time intervals gives $\sum x_{ij} - \sum d_h = \sum R_h$. (This is also clear from Figure 1 with its areas interpretation: Total demand (#) plus total over-allocation (O) equals total allocation.) Hence

$$\text{Ratio} = 100 \times \frac{\sum R_h}{\sum R_h + \sum d_h} = 100 \times \frac{\sum R_h}{\sum x_{ij}} = \frac{\text{Constant}}{\sum x_{ij}}.$$

Therefore Ratio is the normalized performance measure for the GP's first criteria of staff over-allocation. In the extreme (ideal) case of zero over-allocation ($\sum d_h = 0$), it attains its maximum 100% level. It is especially appropriate for comparing different problem instances of different demand patterns generated.

The 25 randomized data sets

Besides this illustration with our project's sample results in Figure 1 above, randomly generated numerical problem instances (with the same model parameters) are reported in Table 1 below to give evidence of the model's robustness. (We allowed the break index $j \in \{1, \cdots, J\}$ – implying a totally flexible break decision interval – for this particular randomization experiment. Further numerical results described later will study the effect on restriction of this break index parameter.) These 25 sets of randomly generated demand requirements are taken from Chu and So [8] where the details of the randomization are provided. For these sets of data (values given in the Appendix A.1), the mean 'utilization' ratio is in fact comparable at 74.83%, with a mean level 4.6 of maximum over allocation. Regarding the columns in Table 1, MinR, MaxR, AvgR and TtlR are, respectively, minimum, maximum, average and total requirements generated; while TtlD, Ratio and MaxD are the total (over-achievement) deviation, the 'utilization' ratio and the maximum deviation from the DGP model outputs for each data set.

Table 1
Results of 25 random problem instances

(i)	Set#	MinR	MaxR	AvgR	TtlR	TtlD	Ratio	MaxD
(1)	9	4	12	7.53	143	17	89.38%	2
(2)	18	4	13	9.84	187	29	86.57%	3
(3)	24	3	13	8.63	164	28	85.42%	3
(4)	7	3	13	9.26	176	32	84.62%	3
(5)	2	3	11	7.47	142	26	84.52%	3
(6)	10	3	13	7.63	145	31	82.39%	3
(7)	17	4	13	8.05	153	39	79.69%	3
(8)	6	3	12	7.42	141	43	76.63%	4
(9)	3	6	13	8.63	164	52	75.93%	4
(10)	25	3	13	7.74	147	53	73.50%	4
(11)	22	4	13	8.53	162	38	81.00%	5
(12)	23	3	13	8.68	165	51	76.39%	5
(13)	8	3	12	7.58	144	48	75.00%	5
(14)	21	3	13	7.84	149	51	74.50%	5
(15)	19	3	13	8.11	154	54	74.04%	5
(16)	11	3	13	7.95	151	57	72.60%	5
(17)	14	3	13	7.26	138	54	71.88%	5
(18)	13	3	13	8.16	155	61	71.76%	5
(19)	1	3	13	8.32	158	66	70.54%	5
(20)	4	3	13	7.42	141	59	70.50%	5
(21)	5	3	13	7.42	141	59	70.50%	5
(22)	15	4	13	9.63	183	81	69.32%	6
(23)	16	3	13	6.68	127	81	61.06%	6
(24)	20	3	13	7.63	145	111	56.64%	8
(25)	12	4	13	7.84	149	115	56.44%	8
	Avg	3.36	12.80	8.05	153.0	53.44	74.83%	4.60

We present the 25 sets of outcomes *re*-ordered in increasing value of D (=MaxD); and for the same MaxD, in decreasing utilization (=Ratio) order. Arranging the 25 sets in this way, it is very noticeable from the last two columns (Ratio, MaxD) that the performance of the computed results (i.e. the Ratio) is highly correlated with the resulting maximum (time-period specific) over allocation (i.e. the MaxD). This ranges from close to 90% utilization with a corresponding maximum over allocation of only 2, to the eighty some percents of MaxD = 3, to the seventy some percents when MaxD = 4 and 5, to the sixty some percents of MaxD = 6, and finally down to only 56% with our largest computed MaxD, being 8 for this 25 sets of randomized sample data. The additional insight gained from this experiment is therefore that the maximum over allocation is actually a rather important performance indicator, even though it is often treated simply as a smoothing measure of secondary priority goal.

Further numerical results (GP)

We further look at the numerical results of the effect on restriction of the important break index parameter j in the decision variables x_{ij}. With the length of a duty $J = 9$, three progressively move restrictive scenarios (as motivated by actual applications) are considered: $j \in [1, 9]$, $j \in [3, 7]$, $j \in [4, 6]$. These same 25 data sets (with their detailed values given in the Appendix A.1) lead to the GP outputs shown in Table 2.

Table 2
GP outputs for the 3 different scenarios

Time/Set	TtlR	GP Output [1,9]			GP Output [3,7]			GP Output [4,6]		
		TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD
1	158	66	70.54%	5	66	70.54%	10	66	70.54%	10
2	142	26	84.52%	3	26	84.52%	8	42	77.17%	8
3	164	52	75.93%	4	52	75.93%	7	52	75.93%	7
4	141	59	70.50%	5	59	70.50%	6	59	70.50%	8
5	141	59	70.50%	5	59	70.50%	8	59	70.50%	8
6	141	43	76.63%	4	43	76.63%	4	43	76.63%	5
7	176	32	84.62%	3	32	84.62%	8	40	81.48%	8
8	144	48	75.00%	5	48	75.00%	8	56	72.00%	8
9	143	17	89.38%	2	17	89.38%	3	17	89.38%	6
10	145	31	82.39%	3	31	82.39%	4	31	82.39%	5
11	151	57	72.60%	5	57	72.60%	7	57	72.60%	8
12	149	115	56.44%	8	115	56.44%	11	115	56.44%	11
13	155	61	71.76%	5	61	71.76%	8	61	71.76%	9
14	138	54	71.88%	5	54	71.88%	7	54	71.88%	8
15	183	81	69.32%	6	81	69.32%	9	81	69.32%	9
16	127	81	61.06%	6	81	61.06%	7	81	61.06%	7
17	153	39	79.69%	3	39	79.69%	7	39	79.69%	7
18	187	29	86.57%	3	29	86.57%	6	45	80.60%	7
19	154	54	74.04%	5	54	74.04%	6	62	71.30%	8
20	145	111	56.64%	8	111	56.64%	10	111	56.64%	11
21	149	51	74.50%	5	51	74.50%	8	59	71.63%	8
22	162	38	81.00%	5	38	81.00%	7	54	75.00%	8
23	165	51	76.39%	5	51	76.39%	9	51	76.39%	9
24	164	28	85.42%	3	28	85.42%	6	36	82.00%	6
25	147	53	73.50%	4	53	73.50%	6	53	73.50%	7
	Avg =		74.83%	4.60		74.83%	7.20		73.45%	7.84
	Min =		56.44%	2		56.44%	3		56.44%	5
	Max =		89.38%	8		89.38%	11		89.38%	11
	Range =		32.94%	6.00		32.94%	8.00		32.94%	6.00
	Std Dev =		8.53%	1.47		8.53%	1.89		7.60%	1.52

As expected, scenario [1,9] performs the best and is naturally taken as the benchmark for performance measures of *low* TtlD (first goal) and *low* MaxD (second goal). Scenario [3,7] comes next and, with a mild nice suprise, attains fully all 25 cases of the TtlD goal, but with now all cases except one (Set 6) of higher MaxD values. Scenario [4,6] reaches only 17 cases (except for Sets 2,7,8,18,19,21,22,24) of the TtlD goal, with all cases of higher MaxD values (and 12 cases compared to scenario [3,7]). Trade-offs between meal break restriction and performance measures, especially the maximum (interval specific) over-allocation level MaxD are clearly evident in this comparison.

2. HEURISTICS ALGORITHMIC APPROACH

Concern on computation robustness of DGP, a GP with *integer* variables, has led us to study the following "best-fitting" type heuristics. Indeed, the solution time for our case application (results in Figure 1) is very fast of less than 1 minute. However, the range of computational times for the 25 random instances in Table 1 is extremely large. For eight of the 25 cases, each takes more than 1.5 hours on a Pantium PC, while the remaining 17 cases average to less than 1 minute. Of the eight "hard" cases, four require manually fixing the single variable MaxD to be integer and solving its LP relaxation instead, due to the fact that each of their ILP times already exceeds our preset 10-hour limit.

(We remark here that the following heuristics has appeared in our electronic proceedings paper of a recent International MOPGP Conference [9]. It is explicitly included here considering the difficulty of readers' gaining access to a paper in electronic proceedings.)

Minimax time-reversible heuristics

Starting at time interval 1 and marching in a time-forward manner, we add each duty sequentially, selecting the break-hour which myopically minimizes its chosen interval's remaining demand (over its L covering intervals). Mathematically, denote

$$r_h \;\equiv\; \text{remaining demand at time } h,\; h = 1, \cdots, H.$$

As we consider a duty starting from time i (when $r_i \geqslant 1$), we add another duty $\triangle x_{ij} = 1$ such that j is chosen as the minimizing index in

$$r_{i+j-1} \;=\; \text{Min}_{k=1,\cdots L}\; r_{i+k-1}.$$

Time intervals are processed from $i = 1$ to $i = H+L-1$ (forward). Note that each chosen break-hour j is locally the time interval with the minimum r_j (as defined above) *and* the maximum d_j, which is the (current value of) over-allocation (when the minimum r_j is zero) at time interval j. (Note that $d_j \times r_j = 0$, or d_j can be positive only when $r_j = 0$.) Hence it is a minimax (and time-forward) greedy heuristics.

An obvious improvement is to process the time intervals starting at time $H + L - 1$ and working in a time-backward manner, i.e. from $i = H + L - 1$ to $i = 1$. This results in a minimax (and time-backward) greedy heuristics.

Combining these two by taking the better performance gives what we call a *Minimax Time-reversible Heuristics*. Its numerical performances for the same set of 25 random problem instances are given in Table 3. It can be seen from Table 3 that 20 out of 25 cases the (GP) optimal 'utilization' ratios are attained (with 12 from Forward alone, and 19 from Backward alone). A perhaps rather surprising further improvement is when we apply randomization to the choice of break-hour j (replacing the above minimax rule). Here this actually gives a higher number of 23 out of 25 cases of optimal ratios. (A side remark on computation: Randomization results naturally vary among different computer runs. In our case, the solution is taken from the best of iterations of 20,000 replicas, taking a total of about 45 minutes on a Pentium PC. Each heuristics trial, in either the minimax or the randomization case, takes negligible amount of computer time.)

Table 3
Heuristics Results of the 25 random problem instances

(0)	(1)	(2)	(3)	(4)
GP Solutions	25	74.83%	25	4.60
Heuristics (Minimax)				
- Forward	12	70.89%	2	7.20
- Backward	19	73.67%	7	6.52
Combined (better of F+B)	20	74.05%	8	5.92
Heuristics (Randomized)				
- Forward	20	74.12%	1	7.04
- Backward	22	74.38%	0	7.12
Combined (better of F+B)	22	74.38%	1	6.76
Complete Heuristics (Minimax+Randomized, with time reversibility)	25	74.83%	9	5.92

Note: Column (0) = Approach
(1) = Number of cases attaining optimal GP's (Maximum) Ratios
(2) = Computed average Ratios
(3) = Number of cases attaining optimal GP's (Minimum) MaxD's
(4) = Computed average MaxD's

For our 25 problem instances, it turns out that together with the final help from randomization, all 25 optimal (GP) ratios are attained by our heuristics. This of course can never be assumed for other different data sets and/or larger scale experiments. Indeed, in terms of the MaxD measure, our complete heuristics can only achieve 9 out of 25 cases of optimal (GP) MaxD's, as shown in Table 3. As expected, the minimax heuristics performs rather better than the randomization (alone) calculations on MaxD. Nevertheless, together all these point to the robust computing benchmark for the DGP optimization model, backed by efficient heuristics as such.

Table 4
Heuristics outputs for the 3 different scenarios

Time/Set	TtlR	The Best Heuristic Output [1,9]			The Best Heuristic Output [3,7]			The Best Heuristic Output [4,6]		
		TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD
1	158	66	70.54%	8	66	70.54%	11	66	70.54%	12
2	142	26	84.52%	3	26	84.52%	8	42	77.17%	12
3	164	52	75.93%	6	52	75.93%	7	52	75.93%	7
4	141	59	70.50%	6	59	70.50%	7	59	70.50%	9
5	141	59	70.50%	7	59	70.50%	8	59	70.50%	8
6	141	43	76.63%	5	43	76.63%	5	43	76.63%	5
7	176	32	84.62%	3	32	84.62%	8	40	81.48%	8
8	144	48	75.00%	6	48	75.00%	8	56	72.00%	9
9	143	17	89.38%	2	17	89.38%	4	17	89.38%	6
10	145	31	82.39%	3	31	82.39%	5	31	82.39%	5
11	151	57	72.60%	7	57	72.60%	9	57	72.60%	12
12	149	115	56.44%	11	115	56.44%	13	115	56.44%	12
13	155	61	71.76%	5	61	71.76%	8	61	71.76%	9
14	138	54	71.88%	6	54	71.88%	7	54	71.88%	9
15	183	81	69.32%	9	81	69.32%	10	81	69.32%	10
16	127	81	61.06%	8	81	61.06%	8	81	61.06%	10
17	153	39	79.69%	5	39	79.69%	7	39	79.69%	7
18	187	29	86.57%	7	29	86.57%	6	53	77.92%	10
19	154	54	74.04%	5	54	74.04%	7	54	74.04%	14
20	145	111	56.64%	11	111	56.64%	12	111	56.64%	14
21	149	51	74.50%	5	51	74.50%	9	59	71.63%	10
22	162	38	81.00%	5	38	81.00%	12	46	77.88%	13
23	165	51	76.39%	7	51	76.39%	9	51	76.39%	10
24	164	28	85.42%	3	28	85.42%	11	52	75.93%	10
25	147	53	73.50%	5	53	73.50%	8	53	73.50%	10
Avg =			74.83%	5.92		74.83%	8.28		73.33%	9.64
Min =			56.44%	2		56.44%	4		56.44%	5
Max =			89.38%	11		89.38%	13		89.38%	14
Range =			32.94%	9.00		32.94%	9.00		32.94%	9.00
Std Dev =			8.53%	2.33		8.53%	2.26		7.36%	2.50
Count =		25		9	25		10	21		8

Further numerical results (heuristics)

Similar to the consideration given to GP computations, we further look at the numerical results of the effect on restriction of the important break index parameter j in the decision variables x_{ij}, for the heuristics. Again, with the length of a duty $J = 9$, three progressively move restrictive scenarios (as motivated by actual applications) are considered: $j \in [1, 9]$, $j \in [3, 7]$, $j \in [4, 6]$. These same 25 input data sets (with their detailed values given in the Appendix A.1) lead to the numerical outputs shown in Table 4. The nice "surprise" here is that Table 4 reads rather consistently similar to Table 2. (See the shaded entries, which indicate common values in both tables.) This consistency is also both in terms of the three different cases attaining their individual different levels of the two goals TtlD and MaxD, *and* the relative progressive degrees of performances (in the goals). For these data sets at least, the heuristics are observed to be also very robust with respect to the key parameter (break index) j of central importance, besides the obvious first goal of TtlD.

3. THE DETAILED COMPARISONS

We have seen from Table 3 the competing performance of the heuristics with respect to the GP solutions for the situation of $j \in [1, 9]$. Perfect performance is scored on the first goal of TtlD, while worse-off level is recorded on the second goal of MaxD (in Table 3). Here Table 5 gives a summary of the detailed comparisons of all three scenarios of $j \in [1, 9]$, $j \in [3, 7]$, $j \in [4, 6]$. It can be seen that out of 25 cases, the (best outputs of the) three heuritics achieve, respectively 25,25,21 cases of their GP counterpart (optimal) solutions on TtlD; and 9,10,8 cases on MaxD. (These are high-lighted as shaded entries in Table 5, which combines Tables 2 and 4.) While the heuristics is confirmed as very competetive on TtlD, it is (intuitively expected to be) much less so on the second goal MaxD. This is an vivid illustration of the superiority of a GP approach, whenever its computation can be completed within an acceptable amount of computer time and resources.

Three additional tables providing all details to the contributing (or relative) performance of forward vs backward as well as potential benefit in randomization are given in the Appendix A.3 for completeness. Table 6 below gives an overall summary concerning these two aspects.

Table 5
Comparison of heuristics vs GP performances for the 3 scenarios

Time/ Set	TtlR	GP Output [1,9]			The Best Heuristic Output [1,9]			GP Output [3,7]			The Best Heuristic Output [3,7]			GP Output [4,6]			The Best Heuristic Output [4,6]		
		TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD
1	158	66	70.54%	5	66	70.54%	8	66	70.54%	10	66	70.54%	11	66	70.54%	10	66	70.54%	12
2	142	26	84.52%	3	26	84.52%	3	26	84.52%	8	26	84.52%	8	42	77.17%	8	42	77.17%	12
3	164	52	75.93%	4	52	75.93%	6	52	75.93%	7	52	75.93%	7	52	75.93%	7	52	75.93%	7
4	141	59	70.50%	5	59	70.50%	6	59	70.50%	6	59	70.50%	7	59	70.50%	8	59	70.50%	9
5	141	59	70.50%	5	59	70.50%	7	59	70.50%	8	59	70.50%	8	59	70.50%	8	59	70.50%	8
6	141	43	76.63%	4	43	76.63%	5	43	76.63%	4	43	76.63%	5	43	76.63%	5	43	76.63%	5
7	176	32	84.62%	3	32	84.62%	3	32	84.62%	8	32	84.62%	8	40	81.48%	8	40	81.48%	8
8	144	48	75.00%	5	48	75.00%	6	48	75.00%	8	48	75.00%	8	56	72.00%	8	56	72.00%	9
9	143	17	89.38%	2	17	89.38%	2	17	89.38%	3	17	89.38%	4	17	89.38%	6	17	89.38%	6
10	145	31	82.39%	3	31	82.39%	3	31	82.39%	4	31	82.39%	5	31	82.39%	5	31	82.39%	5
11	151	57	72.60%	5	57	72.60%	7	57	72.60%	7	57	72.60%	9	57	72.60%	8	57	72.60%	12
12	149	115	56.44%	8	115	56.44%	11	115	56.44%	11	115	56.44%	13	115	56.44%	11	115	56.44%	12
13	155	61	71.76%	5	61	71.76%	5	61	71.76%	8	61	71.76%	8	61	71.76%	9	61	71.76%	9
14	138	54	71.88%	5	54	71.88%	6	54	71.88%	7	54	71.88%	7	54	71.88%	8	54	71.88%	9
15	183	81	69.32%	6	81	69.32%	9	81	69.32%	9	81	69.32%	10	81	69.32%	9	81	69.32%	10
16	127	81	61.06%	6	81	61.06%	8	81	61.06%	7	81	61.06%	8	81	61.06%	7	81	61.06%	10
17	153	39	79.69%	3	39	79.69%	5	39	79.69%	7	39	79.69%	7	39	79.69%	7	39	79.69%	7
18	187	29	86.57%	3	29	86.57%	7	29	86.57%	6	29	86.57%	6	45	80.60%	7	53	77.92%	10
19	154	54	74.04%	5	54	74.04%	5	54	74.04%	6	54	74.04%	7	62	71.30%	8	54	74.04%	14
20	145	111	56.64%	8	111	56.64%	11	111	56.64%	10	111	56.64%	12	111	56.64%	11	111	56.64%	14
21	149	51	74.50%	5	51	74.50%	5	51	74.50%	8	51	74.50%	9	59	71.63%	8	59	71.63%	10
22	162	38	81.00%	5	38	81.00%	5	38	81.00%	7	38	81.00%	12	54	75.00%	8	46	77.88%	13
23	165	51	76.39%	5	51	76.39%	7	51	76.39%	9	51	76.39%	9	51	76.39%	9	51	76.39%	10
24	164	28	85.42%	3	28	85.42%	3	28	85.42%	6	28	85.42%	11	36	82.00%	6	52	75.93%	10
25	147	53	73.50%	4	53	73.50%	5	53	73.50%	6	53	73.50%	8	53	73.50%	7	53	73.50%	10
Avg =			74.83%	4.60		74.83%	5.92		74.83%	7.20		74.83%	8.28		73.45%	7.84		73.33%	9.64
Min =			56.44%	2		56.44%	2		56.44%	3		56.44%	4		56.44%	5		56.44%	5
Max =			89.38%	8		89.38%	11		89.38%	11		89.38%	13		89.38%	11		89.38%	14
Range =			32.94%	6.00		32.94%	9.00		32.94%	8.00		32.94%	9.00		32.94%	6.00		32.94%	9.00
Std Dev =			8.53%	1.47		8.53%	2.33		8.53%	1.89		8.53%	2.26		7.60%	1.52		7.36%	2.50
Count =					25		9				25		10				21		8

Note: 1) S1[1,9] - meal break = randomly assigned with [ESB,LSB]=[1,9]
2) S2[1,9] - meal break = 1st available least demand hour with [ESB,LSB]=[1,9]
3) S1[3,7] - meal break = randomly assigned with [ESB,LSB]=[3,7]
4) S2[3,7] - meal break = 1st available least demand hour with [ESB,LSB]=[3,7]
5) S1[4,6] - meal break = randomly assigned with [ESB,LSB]=[4,6]
6) S2[4,6] - meal break = 1st available least demand hour with [ESB,LSB]=[4,6]
7) Count = no. of cases (out of 25) that the Best Heuristic Output yield the same TtlD and MaxD as that of GP Output.

Table 6
Forward vs backward & randomization aspect for the three scenarios

Scenario	S1:(Forward)/(Backward)	S2:(Forward)/(Backward)	Best of 4
$j \in [\,1, 9\,]$	$(20, 11)\ /\ (22,\ \ 9)$	$(12,\ \ 8)\ /\ (19, 14)$	$(25,\ \ 9)$
$j \in [\,3, 7\,]$	$(17, 14)\ /\ (19, 13)$	$(24, 13)\ /\ (24,\ \ 7)$	$(25, 10)$
$j \in [\,4, 6\,]$	$(16, 13)\ /\ (17, 14)$	$(16, 10)\ /\ (20,\ \ 9)$	$(21,\ \ 8)$

In Table 6, an ordered pair of entries (m, n) represent the numbers of cases for the measures (TtlD, MaxD) computed by the heuristics succeed in attaining the optimal GP results. $S1$ denotes the heuristics randomly assigning the meal break index over $j \in [1, 9]$, $j \in [3, 7]$, $j \in [4, 6]$ — the three tested scenarios indicated in Table 6 as row labels. $S2$ refers to the (heuristic) rule of assigning j as the first available least (residual) demand time interval. (Hence the higher values of (m, n) nearer and up to (25, 25) the better.) While more insight can be gained from the detailed values from the tables in the Appendix A.3, it is already evident from Table 6 that:

– It concurs with intuition that comparable MaxD values are regardless of the direction of forward or backward computation (due to the symmetric or time-reversibility problem nature).

– It is much less expected to see that the computed values of TtlD always benefit from *backward* computations in all six pairs of scenario by time-reversibility settings.

– Assigning meal break by the least demand heuristic rule ($S1$) works well only when the choice of break time interval is wide (i.e. in the case of $j \in [1, 9]$) and is actually worse in the more restrictive cases (i.e. $j \in [3, 7]$ and $j \in [4, 6]$), for the TtlD goal, in comparison to randomized choices ($S2$).

– However, ($S1$) does work better on the whole with respect to the MaxD goal than ($S2$), as might be expected heuristically.

Taking all of the above together, all four combinations of S1:(forward/backword) and S2:(forward/backward) are essential components of our minimax time-reversible heuristics. The complete heuristics fares very well indeed, especially for the case of $j \in [3, 7]$ which is particularly important from an application point of view rendering it the best choice of (meal break) implementation. This choice is further echoed in our final extensive test of 1 000 new cases of similarly randomly generated input demand data. The 1 000 cases were run by the heuristics for the three break time restriction scenarios for a total of 3 000 runs. Out of them, 10 cases for each scenario were solved to optimality by our DGP-GP code. Of these 30 selected cases, only three fail to attain the GP's optimal TtlD performance (Sets 301, 431, 751); and they all belong to the most restrictve scenario of $j \in [4, 6]$. The results are provided in the Appendix A.2, again for the sake of completeness (and additional information on the performance issue on the MaxD goal as well).

CONCLUDING REMARKS

The purpose of this paper is to report by way of DGP modeling and its further extensive computational experience, the advantage of DGP's readily producing improvement over existing manual staff assignment. In this context, we contrast this paper with an earlier work of Chu (2001), where the sole purpose there was to apply DGP and its extended version for a single instance of real data of the airport case study. Exact solutions (as shown in Figure 1 before) were easily computed then for its set of input data.

The integer programming nature of DGP has since then led us from the application to examining much more into DGP as an independent problem, with its more intriguing computational robustness issue. Thus the key contribution of this paper is the construction and the extensive computation experience of the (now proven) effective heuristics, whenever exact GP computations are facing difficulty with certain problem data instances. In short, the model's usefulness to the users is also strengthened by its computational robustness, in both exact solutions and heuristics calculations.

APPENDIX

A.1. The 25 randomly generated input data sets

Time/ Set	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	TtlR
1	7	8	10	6	13	8	12	9	4	13	5	13	6	3	9	11	5	8	8	158
2	8	11	11	9	6	11	7	9	3	5	5	7	7	6	8	10	8	3	8	142
3	13	6	8	11	8	6	12	12	6	6	11	8	9	10	7	8	8	7	8	164
4	9	3	7	3	4	9	3	13	5	5	10	5	5	11	13	9	12	11	4	141
5	13	8	13	8	6	4	6	10	6	3	4	9	3	3	7	9	11	9	9	141
6	10	6	5	4	8	6	11	11	6	7	9	9	3	6	11	4	12	8	5	141
7	3	12	10	12	13	4	5	11	13	11	4	11	11	8	8	13	8	7	12	176
8	8	12	5	9	11	12	6	8	10	5	8	4	10	6	7	6	3	3	11	144
9	5	7	8	7	6	4	9	10	6	5	9	12	8	10	4	11	4	10	8	143
10	8	4	8	4	3	10	7	11	13	8	13	9	11	7	3	12	3	8	3	145
11	9	8	3	11	13	9	4	5	8	8	5	9	9	8	6	12	6	9	9	151
12	11	4	11	8	8	6	5	11	6	9	6	10	8	8	6	11	4	4	13	149
13	13	9	11	10	7	10	4	6	5	11	12	9	7	6	12	7	5	8	3	155
14	12	5	5	5	13	11	11	5	6	6	12	3	4	8	4	6	9	10	3	138
15	13	13	10	12	4	8	7	13	10	9	5	8	13	7	11	10	9	10	11	183
16	9	5	5	5	7	4	11	4	8	9	13	8	3	5	8	4	8	3	8	127
17	4	12	8	6	5	13	11	11	5	8	6	6	10	6	7	8	11	8	8	153
18	7	13	8	13	6	10	10	8	7	9	9	12	4	13	12	11	13	11	11	187
19	5	5	9	8	8	13	11	9	12	8	5	8	7	12	6	4	13	8	3	154
20	8	10	3	3	5	8	5	6	6	13	6	8	4	11	13	6	9	10	11	145
21	11	3	10	12	11	3	5	9	7	5	5	9	8	11	6	13	5	12	4	149
22	9	10	13	8	7	7	12	4	5	5	9	7	10	9	10	12	10	8	7	162
23	12	3	9	11	5	11	3	13	8	7	6	4	10	11	9	13	12	10	8	165
24	5	3	10	10	9	9	9	12	8	3	8	9	7	13	12	4	10	12	11	164
25	4	3	8	7	8	13	9	9	5	7	9	7	11	5	3	12	8	10	9	147

A.2. The 30 selected samples from 1 000 randomized cases

Time/ Set	TtlR	The Best Heuristic Output [1,9]			The Best Heuristic Output [3,7]			The Best Heuristic Output [4,6]		
		TtlD	MaxD(H)	MaxD	TtlD	MaxD(H)	MaxD	TtlD	MaxD(H)	MaxD
1	157	51	5	5	51	9	9	59	10	10
101	154	54	6	4	54	8	8	54	12	8
201	159	41	5	4	41	10	9	57	12	10
301	168	32	4	3	32	9	7	(40<) 48	9	8
431	144	40	4	3	40	9	8	(48<) 56	10	8
511	148	60	7	6	60	6	6	76	10	6
671	145	47	5	5	47	8	8	47	9	9
751	161	31	8	4	31	7	7	(47<) 55	9	8
801	155	45	8	5	45	8	8	53	8	8
901	161	47	5	4	47	7	7	71	10	10
	Remark:	Shaded	cells are	worse-off	heuristics	results.				

A.3. The complete GP and heuristics outputs of the 25 data sets

Time/ Set	TtlR	GP Output [1,9]			S1[1,9] Forward			S1[1,9] Backward			S2[1,9] Forward			S2[1,9] Backward			The Best Heuristic Output [1,9]		
		TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD
1	158	66	70.54%	5	66	70.54%	9	66	70.54%	8	82	65.83%	10	66	70.54%	8	66	70.54%	8
2	142	26	84.52%	3	34	80.68%	5	34	80.68%	5	50	73.96%	6	26	84.52%	3	26	84.52%	3
3	164	52	75.93%	4	52	75.93%	6	52	75.93%	7	76	68.33%	7	60	73.21%	8	52	75.93%	6
4	141	59	70.50%	5	59	70.50%	7	59	70.50%	7	59	70.50%	6	59	70.50%	7	59	70.50%	6
5	141	59	70.50%	5	59	70.50%	7	59	70.50%	7	75	65.28%	7	83	62.95%	9	59	70.50%	7
6	141	43	76.63%	4	43	76.63%	6	43	76.63%	6	43	76.63%	5	43	76.63%	5	43	76.63%	5
7	176	32	84.62%	3	40	81.48%	7	40	81.48%	6	80	68.75%	9	32	84.62%	3	32	84.62%	3
8	144	48	75.00%	5	48	75.00%	6	48	75.00%	6	72	66.67%	8	64	69.23%	5	48	75.00%	6
9	143	17	89.38%	2	25	85.12%	4	25	85.12%	4	33	81.25%	5	17	89.38%	2	17	89.38%	2
10	145	31	82.39%	3	31	82.39%	6	31	82.39%	8	31	82.39%	5	31	82.39%	3	31	82.39%	3
11	151	57	72.60%	5	57	72.60%	7	57	72.60%	7	65	69.91%	7	57	72.60%	7	57	72.60%	7
12	149	115	56.44%	8	115	56.44%	11	115	56.44%	11	115	56.44%	13	115	56.44%	11	115	56.44%	11
13	155	61	71.76%	5	61	71.76%	7	61	71.76%	8	61	71.76%	5	85	64.58%	11	61	71.76%	5
14	138	54	71.88%	5	54	71.88%	6	54	71.88%	7	54	71.88%	6	54	71.88%	6	54	71.88%	6
15	183	81	69.32%	6	81	69.32%	9	81	69.32%	10	89	67.28%	10	81	69.32%	10	81	69.32%	9
16	127	81	61.06%	6	81	61.06%	9	81	61.06%	8	81	61.06%	9	81	61.06%	10	81	61.06%	8
17	153	39	79.69%	3	39	79.69%	5	39	79.69%	6	63	70.83%	6	39	79.69%	5	39	79.69%	5
18	187	29	86.57%	3	37	83.48%	8	29	86.57%	7	77	70.83%	8	37	83.48%	5	29	86.57%	7
19	154	54	74.04%	5	54	74.04%	8	54	74.04%	6	54	74.04%	7	54	74.04%	5	54	74.04%	5
20	145	111	56.64%	8	111	56.64%	11	111	56.64%	11	111	56.64%	12	111	56.64%	13	111	56.64%	11
21	149	51	74.50%	5	51	74.50%	7	51	74.50%	6	51	74.50%	5	51	74.50%	5	51	74.50%	5
22	162	38	81.00%	5	38	81.00%	5	38	81.00%	7	54	75.00%	5	38	81.00%	6	38	81.00%	5
23	165	51	76.39%	5	51	76.39%	7	51	76.39%	7	59	73.66%	6	59	73.66%	8	51	76.39%	7
24	164	28	85.42%	3	36	82.00%	6	28	85.42%	6	28	85.42%	5	28	85.42%	3	28	85.42%	3
25	147	53	73.50%	4	53	73.50%	7	53	73.50%	7	53	73.50%	8	53	73.50%	5	53	73.50%	5
Avg =			74.83%	4.60		74.12%	7.04		74.38%	7.12		70.89%	7.20		73.67%	6.52		74.83%	5.92
Min =			56.44%	2		56.44%	4		56.44%	4		56.44%	5		56.44%	2		56.44%	2
Max =			89.38%	8		85.12%	11		86.57%	11		85.42%	13		89.38%	13		89.38%	11
Range =			32.94%	6.00		28.68%	7.00		30.13%	7.00		28.98%	8.00		32.94%	11.00		32.94%	9.00
Std Dev =			8.53%	1.47		7.62%	1.74		7.97%	1.64		6.95%	2.25		8.90%	2.93		8.53%	2.33
Count1 =							11			9			5			14			
Count2 =																	25		9

Note: 1) S1[1,9] - meal break = randomly assigned with [ESB,LSB]=[1,9]
2) S2[1,9] - meal break = 1st available least demand hour with [ESB,LSB]=[1,9]
3) Count1 = no. of cases (out of 25) that yield the Best Heuristic Output for Ratio as well as MaxD.
(= no. of cases highlighted)
4) Count2 = no. of cases (out of 25) that the Best Heuristic Output yield the same TtlD and MaxD as that of GP Output.
(= no. of cases highlighted)

Time/ Set	TtlR	GP Output [3,7]			S1[3,7]						S2[3,7]						The Best Heuristic Output [3,7]		
					Forward			Backward			Forward			Backward					
		TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD
1	158	**66**	**70.54%**	**10**	66	70.54%	11	66	70.54%	13	66	70.54%	15	66	70.54%	14	**66**	**70.54%**	**11**
2	142	**26**	**84.52%**	**8**	34	80.68%	8	34	80.68%	11	26	84.52%	8	26	84.52%	10	**26**	**84.52%**	**8**
3	164	**52**	**75.93%**	**7**	52	75.93%	7	52	75.93%	9	52	75.93%	7	52	75.93%	12	**52**	**75.93%**	**7**
4	141	**59**	**70.50%**	**6**	59	70.50%	7	59	70.50%	7	59	70.50%	8	59	70.50%	9	**59**	**70.50%**	**7**
5	141	**59**	**70.50%**	**8**	59	70.50%	8	59	70.50%	9	59	70.50%	8	59	70.50%	10	**59**	**70.50%**	**8**
6	141	**43**	**76.63%**	**4**	43	76.63%	5	43	76.63%	8	43	76.63%	5	43	76.63%	9	**43**	**76.63%**	**5**
7	176	**32**	**84.62%**	**8**	40	81.48%	9	40	81.48%	9	32	84.62%	10	32	84.62%	8	**32**	**84.62%**	**8**
8	144	**48**	**75.00%**	**8**	48	75.00%	8	48	75.00%	8	48	75.00%	9	48	75.00%	8	**48**	**75.00%**	**8**
9	143	**17**	**89.38%**	**3**	25	85.12%	4	25	85.12%	4	17	89.38%	4	17	89.38%	4	**17**	**89.38%**	**4**
10	145	**31**	**82.39%**	**4**	31	82.39%	5	31	82.39%	5	31	82.39%	5	31	82.39%	5	**31**	**82.39%**	**5**
11	151	**57**	**72.60%**	**7**	57	72.60%	10	57	72.60%	9	57	72.60%	12	57	72.60%	9	**57**	**72.60%**	**9**
12	149	**115**	**56.44%**	**11**	115	56.44%	15	115	56.44%	13	115	56.44%	18	115	56.44%	16	**115**	**56.44%**	**13**
13	155	**61**	**71.76%**	**8**	61	71.76%	8	61	71.76%	11	61	71.76%	8	61	71.76%	14	**61**	**71.76%**	**8**
14	138	**54**	**71.88%**	**7**	54	71.88%	7	54	71.88%	8	54	71.88%	7	54	71.88%	8	**54**	**71.88%**	**7**
15	183	**81**	**69.32%**	**9**	81	69.32%	13	81	69.32%	10	81	69.32%	15	81	69.32%	12	**81**	**69.32%**	**10**
16	127	**81**	**61.06%**	**7**	81	61.06%	9	81	61.06%	8	81	61.06%	12	81	61.06%	11	**81**	**61.06%**	**8**
17	153	**39**	**79.69%**	**7**	47	76.50%	7	39	79.69%	10	39	79.69%	7	39	79.69%	11	**39**	**79.69%**	**7**
18	187	**29**	**86.57%**	**6**	37	83.48%	6	37	83.48%	9	29	86.57%	6	29	86.57%	9	**29**	**86.57%**	**6**
19	154	**54**	**74.04%**	**6**	54	74.04%	12	54	74.04%	7	54	74.04%	14	54	74.04%	8	**54**	**74.04%**	**7**
20	145	**111**	**56.64%**	**10**	111	56.64%	13	111	56.64%	12	111	56.64%	16	111	56.64%	15	**111**	**56.64%**	**12**
21	149	**51**	**74.50%**	**8**	59	71.63%	10	51	74.50%	9	51	74.50%	9	51	74.50%	9	**51**	**74.50%**	**9**
22	162	**38**	**81.00%**	**7**	46	77.88%	8	46	77.88%	13	46	77.88%	7	38	81.00%	12	**38**	**81.00%**	**12**
23	165	**51**	**76.39%**	**9**	51	76.39%	9	51	76.39%	9	51	76.39%	9	51	76.39%	11	**51**	**76.39%**	**9**
24	164	**28**	**85.42%**	**6**	36	82.00%	10	44	78.85%	8	28	85.42%	11	36	82.00%	9	**28**	**85.42%**	**11**
25	147	**53**	**73.50%**	**6**	53	73.50%	9	53	73.50%	8	53	73.50%	12	53	73.50%	10	**53**	**73.50%**	**8**
	Avg =		74.83%	7.20		73.76%	8.72		73.87%	9.08		74.71%	9.68		74.69%	10.12		74.83%	8.28
	Min =		56.44%	3		56.44%	4		56.44%	4		56.44%	4		56.44%	4		56.44%	4
	Max =		89.38%	11		85.12%	15		85.12%	13		89.38%	18		89.38%	16		89.38%	13
	Range =		32.94%	8.00		28.68%	11.00		28.68%	9.00		32.94%	14.00		32.94%	12.00		32.94%	9.00
	Std Dev =		8.53%	1.89		7.47%	2.65		7.42%	2.25		8.46%	3.73		8.38%	2.82		8.53%	2.26
	Count1 =						10			12			13			7			
	Count2 =																25		10

Note: 1) S1[3,7] - meal break = randomly assigned with [ESB,LSB]=[3,7]
2) S2[3,7] - meal break = 1st available least demand hour with [ESB,LSB]=[3,7]
3) Count1 = no. of cases (out of 25) that yield the Best Heuristic Output for Ratio as well as MaxD.
(= no. of cases highlighted)
4) Count2 = no. of cases (out of 25) that the Best Heuristic Output yield the same TtlD and MaxD as that of GP Output.
(= no. of cases highlighted)

Time/ Set	TtlR	GP Output [4,6]			S1[4,6]						S2[4,6]						The Best Heuristic Output [4,6]		
					Forward			Backward			Forward			Backward					
		TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD	TtlD	Ratio	MaxD
1	158	**66**	**70.54%**	**10**	66	70.54%	14	66	70.54%	12	66	70.54%	15	66	70.54%	14	**66**	**70.54%**	**12**
2	142	**42**	**77.17%**	**8**	50	73.96%	9	42	77.17%	12	50	73.96%	9	50	73.96%	13	**42**	**77.17%**	**12**
3	164	**52**	**75.93%**	**7**	52	75.93%	7	52	75.93%	9	52	75.93%	7	52	75.93%	13	**52**	**75.93%**	**7**
4	141	**59**	**70.50%**	**8**	59	70.50%	9	59	70.50%	9	59	70.50%	9	59	70.50%	10	**59**	**70.50%**	**9**
5	141	**59**	**70.50%**	**8**	59	70.50%	8	59	70.50%	9	59	70.50%	8	59	70.50%	10	**59**	**70.50%**	**8**
6	141	**43**	**76.63%**	**5**	43	76.63%	5	43	76.63%	8	43	76.63%	5	43	76.63%	9	**43**	**76.63%**	**5**
7	176	**40**	**81.48%**	**8**	48	78.57%	12	48	78.57%	9	48	78.57%	12	40	81.48%	8	**40**	**81.48%**	**8**
8	144	**56**	**72.00%**	**8**	56	72.00%	9	56	72.00%	9	56	72.00%	9	56	72.00%	9	**56**	**72.00%**	**9**
9	143	**17**	**89.38%**	**6**	25	85.12%	6	25	85.12%	6	25	85.12%	7	17	89.38%	6	**17**	**89.38%**	**6**
10	145	**31**	**82.39%**	**5**	31	82.39%	7	31	82.39%	5	39	78.80%	9	31	82.39%	5	**31**	**82.39%**	**5**
11	151	**57**	**72.60%**	**8**	65	69.91%	13	73	67.41%	14	65	69.91%	13	57	72.60%	12	**57**	**72.60%**	**12**
12	149	**115**	**56.44%**	**11**	115	56.44%	14	115	56.44%	12	115	56.44%	18	115	56.44%	16	**115**	**56.44%**	**12**
13	155	**61**	**71.76%**	**9**	61	71.76%	9	61	71.76%	14	61	71.76%	9	61	71.76%	14	**61**	**71.76%**	**9**
14	138	**54**	**71.88%**	**8**	54	71.88%	9	54	71.88%	9	54	71.88%	9	54	71.88%	9	**54**	**71.88%**	**9**
15	183	**81**	**69.32%**	**9**	81	69.32%	12	81	69.32%	10	81	69.32%	15	81	69.32%	12	**81**	**69.32%**	**10**
16	127	**81**	**61.06%**	**7**	81	61.06%	10	81	61.06%	10	89	58.80%	13	81	61.06%	12	**81**	**61.06%**	**10**
17	153	**39**	**79.69%**	**7**	47	76.50%	7	39	79.69%	10	39	79.69%	7	39	79.69%	11	**39**	**79.69%**	**7**
18	187	**45**	**80.60%**	**7**	61	75.40%	11	53	77.92%	12	53	77.92%	10	61	75.40%	13	**53**	**77.92%**	**10**
19	154	**62**	**71.30%**	**8**	54	74.04%	15	70	68.75%	9	54	74.04%	14	70	68.75%	10	**54**	**74.04%**	**14**
20	145	**111**	**56.64%**	**11**	111	56.64%	14	111	56.64%	15	111	56.64%	16	111	56.64%	15	**111**	**56.64%**	**14**
21	149	**59**	**71.63%**	**8**	83	64.22%	13	67	68.98%	10	75	66.52%	12	59	71.63%	10	**59**	**71.63%**	**10**
22	162	**54**	**75.00%**	**8**	62	72.32%	9	46	77.88%	13	70	69.83%	10	46	77.88%	13	**46**	**77.88%**	**13**
23	165	**51**	**76.39%**	**9**	59	73.66%	12	59	73.66%	11	67	71.12%	13	51	76.39%	10	**51**	**76.39%**	**10**
24	164	**36**	**82.00%**	**6**	52	75.93%	10	60	73.21%	10	52	75.93%	14	60	73.21%	11	**52**	**75.93%**	**10**
25	147	**53**	**73.50%**	**7**	53	73.50%	10	61	70.67%	10	53	73.50%	12	61	70.67%	11	**53**	**73.50%**	**10**
	Avg =		73.45%	7.84		71.95%	10.16		72.18%	10.28		71.83%	11.00		72.66%	11.04		73.33%	9.64
	Min =		56.44%	5		56.44%	5		56.44%	5		56.44%	5		56.44%	5		56.44%	5
	Max =		89.38%	11		85.12%	15		85.12%	15		85.12%	18		89.38%	16		89.38%	14
	Range =		32.94%	6.00		28.68%	10.00		28.68%	10.00		28.68%	13.00		32.94%	11.00		32.94%	9.00
	Std Dev =		7.60%	1.52		6.76%	2.75		6.99%	2.35		6.84%	3.28		7.30%	2.62		7.36%	2.50
	Count1 =						11			10			10			9			
	Count2 =																21		8

Note: 1) S1[4,6] - meal break = randomly assigned with [ESB,LSB]=[4,6]
2) S2[4,6] - meal break = 1st available least demand hour with [ESB,LSB]=[4,6]
3) Count1 = no. of cases (out of 25) that yield the Best Heuristic Output for Ratio as well as MaxD.
(= no. of cases highlighted)
4) Count2 = no. of cases (out of 25) that the Best Heuristic Output yield the same TtlD and MaxD as that of GP Output.
(= no. of cases highlighted)

REFERENCES

1. Azmat C.S. & Widmer M.: A Case Study of Single Shift Planning and Scheduling under Annualized Hours: A Simple Three-step Approach. ”European Journal of Operational Research” 2004, Vol. 153, pp. 148-175.

2. Bellanti F., Carello G., Della Croce F. & Tadei R.: A Greedy-Based Neighbourhood Search Approach to a Nurse Rostering Problem. ”European Journal of Operational Research” 2004, Vol. 153, pp. 28-40.

3. Bodin L., Golden B., Assad A. & Ball M.: Routing and Scheduling of Vehicles and Crews: The State of the Art. ”Computer and Operations Research” 1983, Vol. 10, pp. 63-211.

4. Burke E. & Petrovic S.: Timetabling and Rostering. ”European Journal of Operational Research” 2004, Vol. 153, pp.1-2.

5. Chu S.C.K.: A Goal Programming Model for Crew Duties Generation. ”Journal of Multi-criteria Decision Analysis” 2001, Vol. 10, pp. 143-151.

6. Chu S.C.K.: Optimization Modeling of Fixed-length Duties. Proceedings of the 32nd International Conference on Computers & Industrial Engineering, Limerick,Ireland,Aug. 2003, pp. 737-742.

7. Chu S.C.K.: Generating, Scheduling and Rostering of Shift Crew-duties: Applications at the Hong Kong International Airport. ”European Journal of Operational Research” (to appear).

8. Chu S.C.K. & So M.M.C.: Generation of Fixed-length Duties by Goal Programming. ”International Journal of Applied Mathematics” 2003, Vol. 13, pp. 9-21.

9. Chu S.C.K. & Yuen C.S.Y.: Generating ShiftCrew-duties. (Electronic) Proceedings of the 6th International Conference on Multi Objective Programming and Goal Programming (MOPGP’04) Hammamet, Tunisia, Apr. 2004, 12pp.

10. Ernst A.T., Jiang H., Krishnamoorthy M. and Sier D.: Staff Scheduling and Rostering: A Review of Applications, Methods and Models. ”European Journal of Operational Research” 2004, Vol. 153, pp. 3-27.

11. Musliu N., Schaerf A. & Slany W.: Local Search for Shift Design. ”European Journal of Operational Research” 2004, Vol. 153, pp. 51-64.

12. Schrage L.: Optimization Modeling with LINGO, 3/e. Lindo Systems Inc. 1999

13. Yuen C.S.Y.: Crew Scheduling and Rostering for Airport Baggage Services: An Optimization Approach. M.Phil Thesis, University of Hong Kong, Hong Kong 2000, 174pp.