International Journal of Operations Research

RESEARCH NOTE

International Journal of Operations Research Vol. 2, No. 2, 89-100 (2005)

A Polynomial Genetic Based Algorithm to Minimize Maximum Lateness in a Two-Stage Flowshop with Setup Times

Ali Allahverdi[1,*], Fawaz S. Al-Anzi[2], and J.N.D. Gupta[3]

[1]Department of Industrial and Management Systems Engineering, College of Engineering and Petroleum, Kuwait University, P.O. Box 5969, Safat, Kuwait

[2]Department of Computer Engineering, College of Engineering and Petroleum, Kuwait University, P.O. Box 5969, Safat, Kuwait

[3]Department of Accounting and Information Systems, College of Administrative Science, The University of Alabama in Huntsville

Abstract—The two-stage flowshop scheduling problem with separate setup times to minimize maximum lateness is addressed in this paper. It is well known that this problem is strongly NP-hard and that there exists at least one optimal solution which is a permutation schedule. A polynomial hybrid genetic based algorithm is proposed to find an approximate solution to this problem. The proposed algorithm is compared with the existing heuristics in the literature. Computational experiments show that the proposed hybrid algorithm significantly outperforms the existing ones. More specifically, the computational complexity of the proposed algorithm and the best existing heuristic is the same as $O(n^3)$ while the average error of the best existing heuristic is 16 times that of proposed algorithm.

Keywords—Scheduling, Flowshop, Lateness, Setup times, Genetic algorithm

1. INTRODUCTION

Consider the following scenario: a set of jobs $N = \{1, 2,..., n\}$ is given to be processed on two machines arranged in series, first on machine 1 and then on machine 2. Associated with each job $i \in N$ are the processing times t_{i1} and t_{i2} at machines 1 and 2, respectively, and a due date d_i. In addition, the processing of job i requires s_{i1} and s_{i2} units of time at machines 1 and 2, respectively, for setup purposes. If job i is completed at time C_i, its lateness L_i is defined as $L_i = C_i - d_i$. It is desired to minimize the maximum lateness $L_{max}=max_{i\in N}(L_i)$. Following Lawler et al (1993), we will denote this problem as a $F2|s_i|L_{max}$ problem where $F2$ denotes two-machine flowshop, s_i implies there are separated sequence independent setup times, and L_{max} indicates that it is desired to minimize maximum lateness. Since the $F2||L_{max}$ problem, a special case of the $F2|s_i|L_{max}$ problem with no setup times, is known be unary NP-hard, it follows that the $F2|s_i|L_{max}$ problem is also unary NP-hard.

The $F2|s_i|L_{max}$ problem was first considered by Dileepan and Sen (1991) who developed a dominance relation and proposed two heuristic algorithms for finding an approximate solution to this problem. Allahverdi and Aldowaisan (1998) also considered the same problem and obtained optimal solutions for special cases. Recently, Allahverdi and Al-Anzi (2002) proposed more heuristic algorithms for the problem and showed that their heuristic algorithms outperform those of Dileepan and Sen. A review of developments in solving flowshop problems involving setup times is provided by Allahverdi, Gupta, and Adowiasan (1999) and Cheng, Gupta, and Wang (2000). A practical application of the $F2|s_i|L_{max}$ problem is discussed by Allahverdi and Al-Anzi (2002).

This paper develops a polynomial hybrid genetic based algorithm to find an approximate solution to the $F2|s_i|L_{max}$ problem. Our goal is to develop a heuristic algorithm that provides better solutions (closer to their optimal value) than those obtained by the best known heuristics that were proposed by Allahverdi and Al-Anzi (2002). The paper proceeds as follows. Section 2 describes the hybrid genetic algorithm to solve the problem. Computational results are provided in Section 3. These computational results show that the proposed hybrid genetic algorithm provides a much better solution than the best existing heuristics that were proposed by Allahverdi and Al-Anzi (2002) while both of them have the same computational complexity. Finally, Section 4 concludes the paper with some directions for future research.

2. THE PROPOSED POLYNOMIAL HYBRID GENETIC BASED ALGORITHM

In this section, we describe the proposed polynomial hybrid genetic based algorithm (GA) for solving the $F2|s_i|L_{max}$ problem. It consists of two phases. In the first phase, a schedule is obtained using a polynomial genetic based algorithm. This schedule is then improved in the

* Corresponding author's email: allahverdi@kuc01.kuniv.edu.kw

second phase using a greedy insertion algorithm. To describe the proposed algorithm, note that any feasible solution of the problem is represented by an ordered set of all jobs (with no repetition), a sequence.

2.1 The genetic based algorithm

Genetic algorithm has been used for the scheduling problems by many researchers including Ruiz and Maroto (2005, 2006), and Tavakkoli-Moghaddam et al. (2005). Our proposed polynomial genetic based algorithm considers a population (*POP*) of given sequences, generated randomly and selects two schedules out of *POP* as parents to produce two offsprings. These two offsprings are produced by swapping subsequences of equal length among two parents. Care must be taken that both offsprings are feasible schedule. To understand this process, consider the following two sequences of X and Y where $X = \{x_1, x_2, \ldots, x_i, \ldots, x_j, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_i, \ldots, y_j, \ldots, y_n\}$. The two segments of $x_i, \ldots, x_n$ and $y_i, \ldots, y_n$ are said to be compatible if they include the same subset of jobs but not necessarily in the same order. Two sequences X and Y are called compatible if they have two compatible segments. The process of generating off springs from a given population is repeated *CP* times.

The process of generating the offsprings is repeated for a given number of generations (*GEN*). Then, y schedules of the population (*POP*) are replaced with the best y schedules from the set of offspring schedules. At the same time, each schedule in the population is mutated with a known probability p. At the end of the given number of generations, a schedule with best value of maximum lateness is accepted as the heuristic solution for the first phase of the proposed algorithm.

The steps of the genetic algorithm of the first phase of the hybrid genetic algorithm are as follows.

Step 1. Initialize a population, *POP*, of random sequences.
Step 2. Compute the L_{max} of each sequence in *POP*.
Step 3. Order the sequences in *POP* according to L_{max} from the best to the worst.
Step 4. Repeat Steps (i) to (v) for *GEN* times
- (i) Repeat steps (a) to (d) for *CP* times
  - (a) Randomly choose two different compatible parents to mutate;
  - (b) Select compatible segments in the two parents;
  - (c) Swap the segments;
  - (d) Save the new sequences in *CHILD* and compute L_{max} of each.
- (ii) Order *CHILD* with respect to L_{max}.
- (iii) Replace the worst y sequences of *POP* with the best y sequences in *CHILD*.
- (iv) Mutate each sequence in *POP* with the probability p.
- (v) Compute L_{max} and order *POP*.

Step 5. Store the best solution from *POP* as π.

It should be noted that the two parents that are used to perform the crossover operation are scanned from left to right. We stop at the earliest position where we can do a swap. That is, the scan process continues until all the positions in both sequences (parents) contain the same set of jobs, not necessarily in the same order. It should also be noted that if y is less than the total number of offsprings, then the remaining offsprings are omitted. On the other hand, if y is greater than the total number of offsprings, we adjust the value of y temporarily to the number of offsprings (this will allow more parents to go into the next generation).

2.2 The insertion algorithm

The solution obtained by the genetic algorithm in the first phase is improved in the second phase by repeated applications of an insertion algorithm. While Dileepan and Sen (1991) used pairwise exchange of jobs to improve the schedule, we propose to use insertion of jobs. For this purpose, we take job r and insert it at as many places in the schedule as possible. The steps of the insertion algorithm of the second stage are as follows:

Step 1. Given an input sequence π of n jobs.
Step 2. Set $r = 1$ and current solution to be empty
Step 3a. Generate r candidate partial sequences by inserting the job at the *r-th* sequence position into each r possible positions of the current solution.
Step 3b.Compute the partial L_{max} for the assigned jobs. Among these candidates, select the one with the least partial L_{max}.
Step 3c.Update the one with the least L_{max} as the current solution.
Step 4.Let $r = r + 1$. If $r < n + 1$, return to Step 3a; otherwise Stop. The best solution is the heuristic solution.

Note that this is the insertion algorithm proposed by Nawaz et al. (1983) but adjusted for our objective function. For example, consider that $r = 1$ and schedule is (1, 2, 3, 4). For $r = 1$, the partial schedule is (1). For $r = 2$, we have (1,2) and (2,1) and suppose the partial L_{max} of (2,1) is smaller, then this partial sequence of (2,1) is chosen in Step 3b. For $r = 3$, the partial sequences of (2,1,3), (2,3,1) and (3,2,1) are evaluated and the best with respect to L_{max} is chosen and it continues.

2.3 The polynomial hybrid genetic based algorithm (GA)

Our proposed hybrid genetic algorithm uses the described polynomial genetic based algorithm in the first phase. The second phase applies the described insertion algorithm repeatedly for a total of q times to improve the schedule obtained in the first phase. Thus, the steps of the proposed hybrid genetic algorithm are as follows:

Phase I. Apply the genetic based algorithm described in section 2.1 to produce an initial solution, π.

Phase II: Apply the insertion algorithm described in section 2.2 on the initial solution π for a total of q times to obtain the final solution.

Careful setting of the parameters for our proposed genetic algorithm is essential to achieve a good performance. This is done experimentally. To do so, various parameter settings were tested with the following ranges: *POP*, *GEN*, and *CP* from n to $5n$ with the increment of n; y from 1/6 to 5/6 with the increment of 1/6; and p from 0.005 to 0.1 with the increment of 0.005; q from 1 to 15 with the increment of 1. After an extensive computational analysis, the parameters are set as given in Table 1 below.

The computational complexity analysis of the first phase is given in Table 2. Therefore, the overall complexity of Phase I is $O(n^3)$. For the Insertion algorithm (Phase II), the number of comparisons required for all the jobs is $1+2+\ldots+n$ since for every job j to be inserted in every possible slot in the partial sequence of size j-1, we have to compute the cost of every candidate partial sequence of size j resulting from inserting the job into every possible slot. This gives the complexity of the insertion algorithm as $O(n^2)$. Hence, the computational complexity of the proposed hybrid genetic algorithm is $O(n^3)$.

Table 1. Parameters of the hybrid genetic algorithm

Parameter	Value
POP	$2n$
GEN	n
CP	$2n$
y	1/3
p	0.035
q	10

Table 2. The computational complexity analysis of the first phase

1.	Initialize a population, *POP*, of random sequences.	$O(n^2)$		
2.	Compute the L_{max} of each sequence in *POP*.	$O(n)$		
3.	Order the sequences in *POP* according to L_{max} from best to worst.	$O(n \log n)$		
4.	Repeat Steps (i) to (v) for *GEN* times	$O(n)$		
	(i) Repeat steps (a) to (d) for *CP* times		$O(n)$	
	(a) Randomly choose two different compatible parents to mutate;			$O(1)$
	(b) Select compatible segments in the two parents;			$O(n)$
	(c) Swap the segments;			$O(n)$
	(d) Save the new sequences in *CHILD* and compute L_{max} of each.			$O(n)$
	(ii) Order CHILD with respect to L_{max}.		$O(n \log n)$	
	(iii) Replace the worst y sequences of POP with the best y sequences in *CHILD*.		$O(n)$	
	(iv) Mutate each sequence in *POP* with the probability p.		$O(n)$	
	(v) Compute L_{max} and order *POP*.		$O(n \log n)$	
5.	Select the best solution from *POP*, and set it to be the current sequence	$O(n)$		

The computational complexities of the heuristics proposed by Dileepan and Sen (1991) (EDD and Johson), those of the heuristics presented by Allahverdi and Al-Anzi (2002) (IH1, IH2, IH3, and IH4), and that of the hybrid genetic based algorithm proposed in this paper (GA) are given in Table 3.

Table 3. The computational complexities of all heuristics

Heuristic	Proposed by	Computational Complexity
EDD	Dileepan and Sen (1991)	$O(n \log n)$
Johnson	Dileepan and Sen (1991)	$O(n \log n)$
IH1	Allahverdi and Al-Anzi (2002)	$O(n^3)$
IH2	Allahverdi and Al-Anzi (2002)	$O(n^3)$
IH3	Allahverdi and Al-Anzi (2002)	$O(n^3)$
IH4	Allahverdi and Al-Anzi (2002)	$O(n^3)$
GA	This paper	$O(n^3)$

3. COMPUTATIONAL EXPERIMENTS

The proposed polynomial hybrid genetic based algorithm (GA) along with the heuristic algorithms developed by Dileepan and Sen (1991) (EDD and Johnson) and by Allahverdi and Al-Anzi (2002) (IH1, IH2, IH3, and IH4) were implemented in C on a Sun Sparc 20, and evaluated with respect to average percentage error, standard deviation of the error, and the number of times yielding the best solution.

Problem data were randomly generated from a uniform distribution with processing times from [1; 100]. In the scheduling literature, most researchers have used this distribution in their experimentation, e.g., Wang et al. (1997), Pan and Chen (1997). The reason for using a uniform distribution with a wide range is that the variance of this distribution is large and if a heuristic performs well with such a distribution, it is likely to perform well with other distributions. The setup times are generated from [0; 100k] as described in Allahverdi and Al-Anzi (2002). The parameter k is the expected ratio of setup to processing times. We generated due dates from a discrete uniform distribution in a range (Px; Pz) following the method by Potts and Van Wassenhove (1982) and Kim (1993) where P is set to the sum of the setup plus processing times of all the jobs on the second machine. The parameters x and y are defined as: $x = (1 - T - R/2)$, $z = (1 - T + R/2)$ where R is called due date range whereas T is called tardiness factor.

Problem data were generated for different combinations of k, T and R values (k = 0.2, 0.8; T = 0.3, 0.6, 1.0; R = 0.3, 0.6,1.0). The experiments are performed for the number of jobs of 30, 40, 50, 60, 70, and 80. We compare the performance of the heuristics for 50 replicates using three measures: average percentage error (Avg), standard deviation (Std), and the number of times the best solution is obtained (NOS). The percentage error is defined as *100* (L_{max} of the heuristic − L_{max} of the best solution)/ (L_{max} of the worst solution − L_{max} of the best solution).*

Tables 4-9 show performance of all of the seven heuristics from n = 30 to n = 80 in the increment of 10 for all combinations of k, T, and R. The overall Avg., Std, and NOS over all n are summarized in Figures 1, 2, and 3, respectively. It is clear that the proposed GA performs much better than all the existing heuristics (Johnson was removed from some of the figures in order to make the figures more readable since it has high error value). More specifically, the overall average percentage errors of EDD, Johnson, IH1, IH2, IH3, IH4, and GA are 28.4156, 98.9550, 9.5582, 14.1575, 3.9356, 8.6509, and 0.2391, respectively. Among the existing heuristics IH3 and IH4 are the best ones. The error of the best existing one (IH3) is more than 16 times that of the proposed GA. Moreover, both IH3 and GA have the same computational complexity. Therefore, GA is superior to the existing heuristics.

Figure 4 shows the average error of the heuristics for the two values of k, which is the ratio of setup to processing times. The figure shows that when the ratio of setup to processing times is small, the error is more for all the heuristics. Figure 5 shows the effect of T on the performance of the heuristics. It seems that the T value does not affect the performance of the heuristics. The effect of R on the performance of the heuristics is shown in Figure 6. The figure shows that as R increases the error increases in general for all heuristics except for IH3. Finally, Figure 7 shows the effect of n on the error. Initially, as n increases the error decreases for all existing heuristics to a level, then it remains unchanged except EDD.

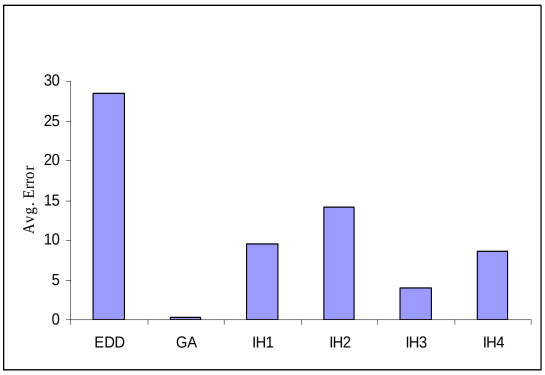

Figure 1. Overall Avg. error.

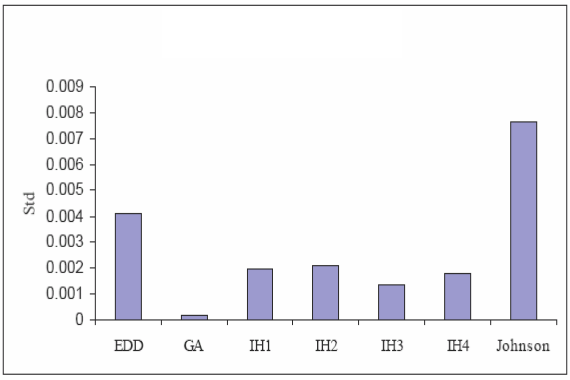

Figure 2. Overall Avg. std.

Figure 3. Overall Avg. percentage of NOS.

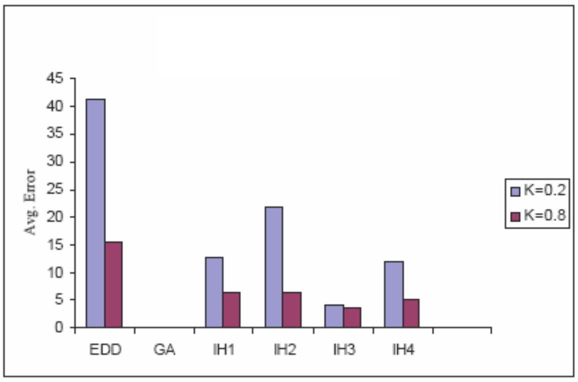

Figure 4. The effect of *k*.

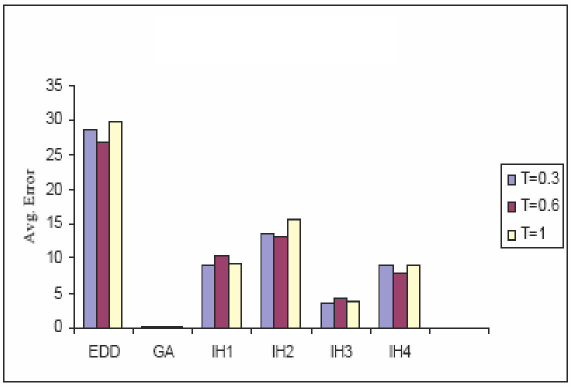

Figure 5. The effect of *T*.

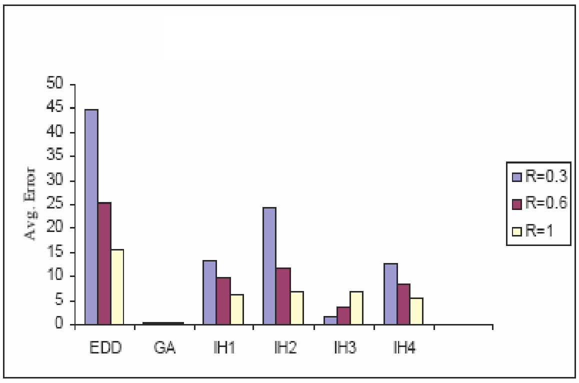

Figure 6. The effect of *R*.

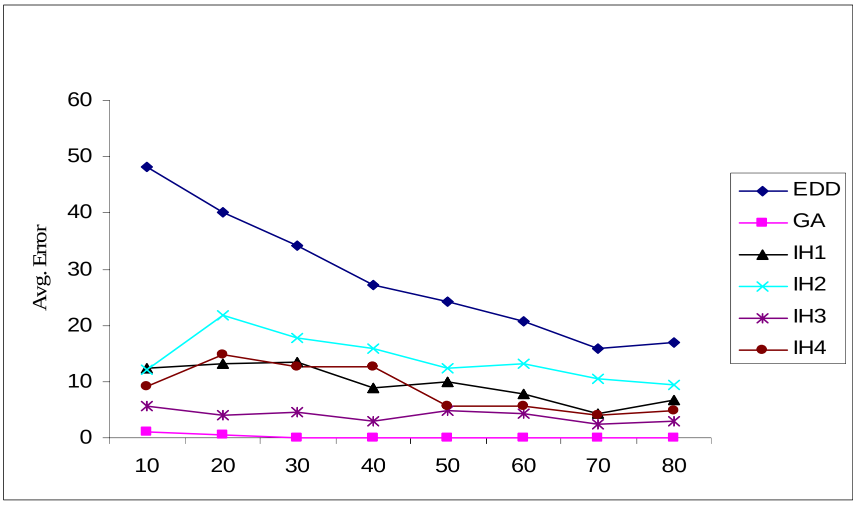

Figure 7. The effect of *n*.

4. CONCLUSIONS

A two-stage hybrid genetic based algorithm is proposed to solve the two-machine flowshop scheduling problem of minimizing maximum lateness with separate and sequence independent setup times. The performance of the proposed hybrid genetic based algorithm and those of the existing heuristic algorithms in the literature are compared. The computational complexity of the proposed genetic algorithm is the same as that of the best existing heuristic in the literature while the error of the best existing heuristic is about 16 times that of the new proposed hybrid genetic algorithm. This shows the superiority of the proposed algorithm over the existing ones.

Since the proposed algorithm performs so well for the considered problem, it can be tested for the same problem but with different objective functions. Moreover, the proposed algorithm can be assessed for multi-machine flowshop problems containing more than two stages. In general, it can be tested for all other related scheduling problems.

REFERENCES

1. Allahverdi, A., and Aldowiasan, T. (1998). Job lateness in flowshops with setup and removal times separated. *Journal of Operational Research Society*, 49: 1001-1006.
2. Allahverdi, A., and Al-Anzi, F. (2002). Using two-machine flowshop with maximum lateness objective to model multimedia data objects scheduling problems for WWW applications. *Computers and Operations Research*, 29: 971-994.
3. Allahverdi, A., Gupta, J.N.D., and Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *OMEGA The International Journal of Management Sciences,* 27: 219-239.
4. Cheng, T.C.E., Gupta, J.N.D., and Wang, G. (2000). A review of flowshop scheduling research with setup times. *Production and Operations Management*, 9: 283-302.
5. Dileepan, P. and Sen, T. (1991). Job lateness in a two-machine flowshop with setup times separated. *Computers and Operations Research*, 18: 549-556.
6. Kim, Y.D. (1993). A new branch and bound algorithm for minimizing mean tardiness in two-machine flowshops. *Computers and Operations Research*, 20: 391-401.
7. Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., and Shmoys, D.B. (1993). *Sequencing and Scheduling: Algorithms and Complexity*, in S. C. Graves et al. (eds), Handbooks in Operations Research and Management Science, vol. 4, Elsevier Science Publishers, Amsterdam, Holland, pp. 445-552.
8. Nawaz, M., Enscore, E., and Ham, I. (1983). A heuristic algorithm for the m-machine, *n*-job flowshop sequencing problem. *OMEGA The International Journal of Management Sciences*, 11: 91-95.
9. Pan, C.H., and Chen, J.S., (1997). Scheduling alternative operations in two-machine flow-shops. *Journal of the Operational Research Society*, 48: 533-540.
10. Potts, C.N. and Van Wassenhove, L.N. (1982). A decomposition algorithm for the single machine total tardiness problem. *Operations Research Letters*, 1: 177-181.
11. Ruiz, R. and Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, 165: 479-494.
12. Ruiz, R., and Maroto, C. (2006). A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research*, 169: 781-800.
13. Tavakkoli-Moghaddam, R., Aryanezhad, M.B., Safaei, N., and Azaron, A. (2005). Solving a dynamic cell formation problem using metaheuristics. *Applied Mathematics and Computation*, 170: 761-780.
14. Wang, M.Y., Sethi, S.P., and Van De Velde, S.L. (1997). Minimizing makespan in a class of reentrant shops. *Operations Research*, 45: 702-712.

Table 4. Computational Results for the Heuristics (n=30)

			GA			EDD			IH1			IH2			Johnson			IH3			IH4		
k	T	R	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS
0.2	0.3	0.3	0.000	0.00000	100	85.455	0.00538	50	7.073	0.00058	80	25.431	0.00147	63	100.000	0.00315	13	0.523	0.00010	96	21.556	0.00163	73
0.2	0.3	0.6	0.000	0.00000	100	25.765	0.00306	63	11.098	0.00181	80	16.016	0.00218	73	100.000	0.00800	26	5.774	0.00185	93	17.960	0.00272	76
0.2	0.3	1	0.000	0.00000	100	13.750	0.00424	56	5.880	0.00252	70	7.434	0.00342	70	100.000	0.01838	20	10.670	0.00557	83	6.869	0.00277	63
0.2	0.6	0.3	0.000	0.00000	100	88.000	0.00431	60	36.918	0.00233	66	45.311	0.00213	56	100.000	0.00433	46	1.180	0.00020	96	22.689	0.00165	76
0.2	0.6	0.6	1.541	0.00041	93	38.809	0.00349	56	17.697	0.00234	73	27.067	0.00344	70	100.000	0.00723	23	6.982	0.00150	90	21.838	0.00274	73
0.2	0.6	1	0.000	0.00000	100	30.019	0.00352	40	21.429	0.00317	46	20.102	0.00323	53	100.000	0.01398	23	9.035	0.00335	83	13.872	0.00307	70
0.2	1	0.3	0.000	0.00000	100	83.012	0.00441	60	12.250	0.00109	83	64.907	0.00421	66	100.000	0.00344	36	0.000	0.00000	100	12.623	0.00134	86
0.2	1	0.6	0.000	0.00000	100	61.923	0.00592	53	15.856	0.00242	76	34.949	0.00390	60	100.000	0.00736	26	6.085	0.00140	86	31.078	0.00444	76
0.2	1	1	0.000	0.00000	100	38.258	0.00609	43	24.118	0.00579	66	21.383	0.00431	63	100.000	0.00979	20	10.431	0.00308	76	17.952	0.00417	56
0.8	0.3	0.3	0.000	0.00000	100	25.038	0.00271	70	12.956	0.00097	76	10.003	0.00068	76	100.000	0.00355	33	0.000	0.00000	100	10.817	0.00118	86
0.8	0.3	0.6	0.000	0.00000	100	11.384	0.00198	76	8.179	0.00129	80	1.585	0.00037	93	100.000	0.00590	10	1.566	0.00030	90	1.913	0.00029	86
0.8	0.3	1	0.000	0.00000	100	13.145	0.00579	73	5.349	0.00351	80	4.605	0.00310	76	100.000	0.01382	10	5.679	0.00320	83	6.168	0.00298	76
0.8	0.6	0.3	0.000	0.00000	100	24.742	0.00207	86	23.431	0.00207	90	5.044	0.00063	93	100.000	0.00301	36	3.415	0.00047	96	14.972	0.00139	90
0.8	0.6	0.6	0.000	0.00000	100	8.574	0.00216	83	8.189	0.00216	86	2.119	0.00047	90	100.000	0.00555	13	0.578	0.00020	96	3.452	0.00055	83
0.8	0.6	1	0.000	0.00000	100	13.844	0.00356	76	7.244	0.00258	86	4.481	0.00180	83	100.000	0.00884	6	9.973	0.00323	76	6.957	0.00196	76
0.8	1	0.3	0.506	0.00007	96	28.415	0.00187	76	7.884	0.00075	86	20.784	0.00150	83	100.000	0.00248	30	1.560	0.00015	93	11.425	0.00121	86
0.8	1	0.6	0.000	0.00000	100	13.746	0.00239	70	7.054	0.00166	83	8.630	0.00161	76	100.000	0.00817	33	1.882	0.00056	90	3.848	0.00101	90
0.8	1	1	0.000	0.00000	100	9.473	0.00329	76	7.489	0.00307	80	0.772	0.00038	93	100.000	0.01032	6	7.489	0.00439	93	1.691	0.00100	90
Overall Average			0.11374	0.00003	99.4	34.0751	0.00368	64.8	13.3386	0.00223	77.1	17.8123	0.00216	74.3	100.000	0.00763	22.8	4.60126	0.00164	90.0	12.6488	0.00200	78.4

Table 5. Computational results for the heuristics (n=40)

			GA			EDD			IH1			IH2			Johnson			IH3			IH4		
k	T	R	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS
0.2	0.3	0.3	0.000	0.00000	100	63.225	0.00247	56	23.451	0.00158	80	49.459	0.00254	63	100.000	0.00252	26	1.327	0.00015	96	42.920	0.00188	66
0.2	0.3	0.6	0.000	0.00000	100	60.640	0.00664	36	15.586	0.00354	63	21.094	0.00381	60	100.000	0.00800	20	1.371	0.00027	83	21.514	0.00381	50
0.2	0.3	1	0.000	0.00000	100	19.041	0.00375	63	5.769	0.00247	86	10.949	0.00263	63	100.000	0.01085	23	3.456	0.00114	83	5.180	0.00182	86
0.2	0.6	0.3	0.000	0.00000	100	39.813	0.00237	66	11.139	0.00088	83	16.457	0.00171	90	100.000	0.00304	30	3.809	0.00034	90	13.654	0.00089	76
0.2	0.6	0.6	0.000	0.00000	100	55.553	0.00381	56	22.030	0.00171	70	29.416	0.00211	63	100.000	0.00350	23	0.310	0.00007	96	14.928	0.00143	73
0.2	0.6	1	0.221	0.00016	96	15.457	0.00287	46	3.912	0.00115	76	9.764	0.00212	56	100.000	0.01157	6	8.607	0.00328	76	13.973	0.00433	50
0.2	1	0.3	0.000	0.00000	100	89.448	0.00330	40	13.429	0.00103	76	39.968	0.00174	53	100.000	0.00249	23	3.357	0.00032	93	46.443	0.00229	53
0.2	1	0.6	0.384	0.00008	96	54.871	0.00348	43	14.395	0.00142	70	33.930	0.00239	60	100.000	0.00387	10	0.000	0.00000	100	21.887	0.00143	56
0.2	1	1	0.125	0.00007	96	13.132	0.00256	63	5.670	0.00164	76	8.812	0.00190	70	100.000	0.00966	23	7.490	0.00245	86	7.969	0.00170	70
0.8	0.3	0.3	0.000	0.00000	100	5.479	0.00071	90	4.257	0.00070	96	8.614	0.00093	90	100.000	0.00296	20	0.297	0.00005	96	0.627	0.00011	96
0.8	0.3	0.6	0.000	0.00000	100	17.725	0.00182	73	8.202	0.00091	80	12.280	0.00114	70	100.000	0.00472	30	0.834	0.00014	93	9.337	0.00113	83
0.8	0.3	1	0.000	0.00000	100	1.983	0.00060	86	0.000	0.00000	100	3.215	0.00115	90	100.000	0.00970	16	9.425	0.00265	73	5.970	0.00147	80
0.8	0.6	0.3	0.000	0.00000	100	13.737	0.00088	80	13.339	0.00084	80	17.403	0.00099	76	100.000	0.00224	26	0.398	0.00005	96	9.982	0.00073	83
0.8	0.6	0.6	0.000	0.00000	100	15.593	0.00223	80	8.229	0.00119	83	12.240	0.00198	83	100.000	0.00597	20	1.070	0.00028	96	2.674	0.00066	93
0.8	0.6	1	0.000	0.00000	100	10.461	0.00275	80	5.882	0.00155	83	1.078	0.00048	90	100.000	0.00959	20	8.439	0.00280	83	4.435	0.00123	86
0.8	1	0.3	0.000	0.00000	100	7.289	0.00085	90	0.000	0.00000	100	7.331	0.00063	83	100.000	0.00258	26	0.000	0.00000	100	3.100	0.00036	93
0.8	1	0.6	0.000	0.00000	100	4.211	0.00060	80	3.113	0.00057	86	3.725	0.00059	83	100.000	0.00422	10	0.000	0.00000	100	2.411	0.00056	90
0.8	1	1	0.000	0.00000	100	2.906	0.00096	80	1.068	0.00036	86	1.943	0.00055	80	100.000	0.00957	16	4.167	0.00157	83	1.996	0.00068	83
Overall Average			0.04049	0.00002	99.3	27.2535	0.00237	67.1	8.8596	0.00120	81.9	15.9822	0.00163	73.5	100.000	0.00595	20.4	3.0198	0.00086	90.2	12.722	0.00147	75.9

Table 6. Computational results for the heuristics (n=50)

k	T	R	GA			EDD			IH1			IH2			Johnson			IH3			IH4		
			Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS
0.2	0.3	0.3	0.000	0.00000	100	57.137	0.00251	56	12.038	0.00087	83	21.711	0.00104	70	100.000	0.00198	20	0.000	0.00000	100	1.376	0.00013	93
0.2	0.3	0.6	0.000	0.00000	100	28.667	0.00295	63	11.353	0.00173	83	11.451	0.00130	73	100.000	0.00608	23	0.451	0.00009	93	8.922	0.00131	83
0.2	0.3	1	0.000	0.00000	100	21.176	0.00299	46	8.490	0.00194	63	8.654	0.00183	66	100.000	0.00900	13	9.992	0.00206	73	8.845	0.00206	63
0.2	0.6	0.3	0.000	0.00000	100	42.562	0.00215	53	25.302	0.00126	66	21.423	0.00127	66	100.000	0.00258	16	0.854	0.00013	96	5.196	0.00048	86
0.2	0.6	0.6	0.972	0.00019	96	26.007	0.00169	60	16.560	0.00158	80	20.922	0.00159	66	100.000	0.00421	30	4.890	0.00067	93	11.475	0.00084	73
0.2	0.6	1	0.107	0.00006	96	17.213	0.00221	43	4.949	0.00131	73	5.337	0.00098	63	100.000	0.00854	6	14.176	0.00329	80	4.648	0.00124	76
0.2	1	0.3	0.000	0.00000	100	100.000	0.00253	60	16.828	0.00072	80	70.680	0.00192	70	88.544	0.00182	36	0.000	0.00000	100	17.346	0.00086	83
0.2	1	0.6	0.000	0.00000	100	28.700	0.00224	60	14.755	0.00160	73	13.688	0.00119	73	100.000	0.00479	33	0.470	0.00009	93	9.972	0.00098	76
0.2	1	1	0.000	0.00000	100	18.400	0.00182	40	11.153	0.00122	60	9.395	0.00109	60	100.000	0.00762	10	17.675	0.00336	76	4.872	0.00085	80
0.8	0.3	0.3	0.000	0.00000	100	13.062	0.00134	90	1.420	0.00012	93	1.420	0.00012	93	100.000	0.00216	16	0.757	0.00009	96	1.088	0.00009	93
0.8	0.3	0.6	0.000	0.00000	100	7.854	0.00093	73	6.839	0.00092	80	2.413	0.00057	86	100.000	0.00473	20	3.178	0.00078	90	4.842	0.00093	83
0.8	0.3	1	0.000	0.00000	100	8.548	0.00142	76	1.596	0.00048	93	0.212	0.00008	96	100.000	0.00731	20	2.797	0.00100	93	3.348	0.00091	86
0.8	0.6	0.3	0.000	0.00000	100	20.879	0.00130	80	13.880	0.00093	83	10.070	0.00072	83	100.000	0.00256	10	1.128	0.00016	96	5.288	0.00046	86
0.8	0.6	0.6	0.422	0.00011	96	5.529	0.00053	70	4.624	0.00050	73	2.533	0.00038	86	100.000	0.00438	16	8.243	0.00115	86	0.302	0.00006	93
0.8	0.6	1	0.000	0.00000	100	5.090	0.00106	83	1.487	0.00037	90	3.295	0.00086	83	100.000	0.00495	3	6.322	0.00136	86	3.054	0.00071	83
0.8	1	0.3	0.000	0.00000	100	7.528	0.00034	90	7.528	0.00034	90	9.581	0.00033	86	100.000	0.00146	40	0.000	0.00000	100	0.000	0.00000	100
0.8	1	0.6	0.000	0.00000	100	14.300	0.00176	76	14.130	0.00176	80	5.081	0.00066	83	100.000	0.00433	26	6.025	0.00095	86	4.283	0.00050	83
0.8	1	1	0.000	0.00000	100	12.151	0.00236	73	6.602	0.00208	90	4.626	0.00165	86	100.000	0.00773	13	9.678	0.00255	80	5.099	0.00144	83
Overall Average			0.08341	0.00002	99.3	24.1556	0.00178	66.2	9.97417	0.00110	79.6	12.3606	0.00098	77.2	99.3635	0.00479	19.5	4.81314	0.00098	89.8	5.55310	0.00077	83.5

Table 7. Computational results for the heuristics (n=60)

			GA			EDD			IH1			IH2			Johnson			IH3			IH4		
k	T	R	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS
0.2	0.3	0.3	0.619	0.00006	96	49.319	0.00139	60	9.653	0.00038	76	38.738	0.00132	63	100.000	0.00169	23	4.208	0.00024	90	18.626	0.00079	73
0.2	0.3	0.6	0.000	0.00000	100	30.847	0.00201	53	16.107	0.00146	70	19.400	0.00171	66	100.000	0.00380	16	2.419	0.00054	93	2.397	0.00050	93
0.2	0.3	1	0.000	0.00000	100	33.382	0.00381	56	12.929	0.00246	73	5.643	0.00067	70	100.000	0.00596	16	2.940	0.00061	90	2.630	0.00052	86
0.2	0.6	0.3	0.000	0.00000	100	57.767	0.00250	63	5.453	0.00041	93	20.782	0.00087	73	100.000	0.00202	26	0.000	0.00000	100	8.025	0.00060	90
0.2	0.6	0.6	1.103	0.00016	96	27.868	0.00161	60	18.419	0.00108	66	16.801	0.00113	73	100.000	0.00283	30	2.426	0.00032	93	8.419	0.00078	90
0.2	0.6	1	0.000	0.00000	100	7.519	0.00125	53	6.224	0.00112	60	3.379	0.00076	70	100.000	0.00762	10	17.573	0.00392	60	6.721	0.00158	73
0.2	1	0.3	0.687	0.00005	96	36.336	0.00131	73	10.153	0.00033	80	46.107	0.00127	66	100.000	0.00148	33	0.000	0.00000	100	13.130	0.00038	76
0.2	1	0.6	0.000	0.00000	100	34.635	0.00174	53	16.583	0.00131	73	31.196	0.00160	53	100.000	0.00312	26	1.039	0.00011	93	6.411	0.00040	76
0.2	1	1	0.000	0.00000	100	5.498	0.00079	66	4.443	0.00073	70	6.566	0.00111	66	100.000	0.00691	10	15.092	0.00231	70	2.470	0.00067	83
0.8	0.3	0.3	0.000	0.00000	100	26.601	0.00142	76	14.062	0.00083	83	14.330	0.00084	83	100.000	0.00147	10	0.000	0.00000	100	6.225	0.00046	86
0.8	0.3	0.6	0.000	0.00000	100	12.560	0.00114	73	3.972	0.00049	86	6.935	0.00076	80	100.000	0.00285	10	1.768	0.00042	96	5.304	0.00059	83
0.8	0.3	1	0.000	0.00000	100	9.293	0.00165	76	3.098	0.00080	86	3.614	0.00080	80	100.000	0.00589	16	4.389	0.00108	83	3.824	0.00060	83
0.8	0.6	0.3	0.000	0.00000	100	10.547	0.00062	80	3.751	0.00031	90	4.722	0.00026	80	100.000	0.00175	10	3.089	0.00039	96	1.103	0.00011	93
0.8	0.6	0.6	0.000	0.00000	100	4.676	0.00070	86	4.225	0.00063	86	2.089	0.00045	93	100.000	0.00429	16	2.753	0.00038	83	1.044	0.00016	90
0.8	0.6	1	0.000	0.00000	100	1.789	0.00046	83	1.453	0.00045	86	3.588	0.00063	73	100.000	0.00661	3	5.567	0.00141	80	0.727	0.00029	90
0.8	1	0.3	0.000	0.00000	100	3.836	0.00022	86	0.918	0.00007	93	5.835	0.00038	80	100.000	0.00188	13	5.943	0.00060	96	7.455	0.00034	76
0.8	1	0.6	0.000	0.00000	100	11.420	0.00164	73	5.610	0.00101	76	5.447	0.00088	80	100.000	0.00410	6	3.050	0.00045	80	3.813	0.00070	80
0.8	1	1	0.000	0.00000	100	9.138	0.00237	83	5.269	0.00116	86	4.021	0.00085	83	100.000	0.00616	3	3.993	0.00111	90	1.719	0.00052	86
Overall Average			0.13382	0.00001	99.3	20.7239	0.00148	69.6	7.90680	0.00083	79.6	13.2884	0.00090	74.0	100.000	0.00391	15.4	4.23609	0.00077	88.5	5.55795	0.00055	83.7

Table 8. Computational results for the heuristics (n=70)

			GA			EDD			IH1			IH2			Johnson			IH3			IH4		
k	T	R	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS
0.2	0.3	0.3	0.000	0.00000	100	55.886	0.00151	63	10.696	0.00048	86	43.291	0.00128	66	100.000	0.00152	23	0.000	0.00000	100	10.506	0.00047	86
0.2	0.3	0.6	0.000	0.00000	100	10.078	0.00089	66	2.242	0.00032	83	11.410	0.00101	70	100.000	0.00337	10	1.021	0.00012	86	5.261	0.00066	83
0.2	0.3	1	0.000	0.00000	100	20.055	0.00222	53	7.547	0.00115	76	13.142	0.00127	63	100.000	0.00582	13	6.636	0.00115	83	4.652	0.00069	80
0.2	0.6	0.3	0.000	0.00000	100	31.081	0.00106	73	7.400	0.00041	90	20.142	0.00100	83	100.000	0.00177	26	5.084	0.00033	93	3.089	0.00026	96
0.2	0.6	0.6	0.000	0.00000	100	21.454	0.00119	66	3.846	0.00031	83	3.423	0.00028	83	100.000	0.00297	23	2.435	0.00024	90	4.058	0.00041	86
0.2	0.6	1	0.000	0.00000	100	6.546	0.00152	80	3.527	0.00115	86	1.806	0.00030	76	100.000	0.00664	6	0.917	0.00035	96	2.878	0.00057	76
0.2	1	0.3	0.000	0.00000	100	38.854	0.00134	63	4.777	0.00025	83	29.512	0.00117	70	100.000	0.00197	33	0.000	0.00000	100	5.892	0.00032	83
0.2	1	0.6	0.000	0.00000	100	25.462	0.00179	56	3.749	0.00045	86	17.348	0.00106	60	100.000	0.00323	13	1.315	0.00016	90	1.567	0.00016	86
0.2	1	1	0.000	0.00000	100	8.936	0.00121	70	4.116	0.00073	83	9.108	0.00114	70	100.000	0.00601	23	4.914	0.00087	80	4.460	0.00080	83
0.8	0.3	0.3	0.000	0.00000	100	13.345	0.00062	83	9.382	0.00051	86	10.315	0.00055	90	100.000	0.00172	16	0.932	0.00006	93	8.625	0.00047	80
0.8	0.3	0.6	0.000	0.00000	100	20.403	0.00176	70	4.819	0.00038	76	4.026	0.00037	83	100.000	0.00323	23	6.709	0.00072	86	4.117	0.00034	83
0.8	0.3	1	0.000	0.00000	100	2.936	0.00056	86	2.092	0.00051	93	3.325	0.00057	80	100.000	0.00426	6	2.806	0.00060	90	0.422	0.00008	90
0.8	0.6	0.3	0.000	0.00000	100	8.176	0.00041	83	4.631	0.00025	86	16.066	0.00079	76	100.000	0.00149	20	0.000	0.00000	100	1.601	0.00016	96
0.8	0.6	0.6	0.000	0.00000	100	3.985	0.00065	86	1.245	0.00025	93	0.453	0.00008	93	100.000	0.00342	10	0.000	0.00000	100	4.935	0.00087	86
0.8	0.6	1	0.000	0.00000	100	3.210	0.00063	86	0.000	0.00000	100	1.020	0.00018	83	100.000	0.00537	3	3.840	0.00077	76	1.080	0.00020	83
0.8	1	0.3	0.000	0.00000	100	2.505	0.00022	93	0.511	0.00006	96	2.301	0.00022	93	100.000	0.00189	23	0.818	0.00006	93	6.135	0.00044	86
0.8	1	0.6	0.000	0.00000	100	12.311	0.00136	86	3.907	0.00058	96	0.958	0.00012	93	100.000	0.00308	13	3.539	0.00030	86	0.774	0.00011	96
0.8	1	1	0.000	0.00000	100	2.187	0.00034	76	1.005	0.00017	83	1.227	0.00018	80	100.000	0.00600	10	1.508	0.00031	86	0.473	0.00013	93
Overall Average			0.00000	0.00000	100	15.9672	0.00107	74.4	4.19412	0.00044	86.9	10.4928	0.00064	78.4	100.000	0.00354	16.3	2.35964	0.00034	90.4	3.91802	0.00040	86.2

Table 9. Computational results for the heuristics (n=80)

			GA			EDD			IH1			IH2			Johnson			IH3			IH4		
k	T	R	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS	Avg	Std	NOS
0.2	0.3	0.3	0.000	0.00000	100	32.184	0.00098	66	16.455	0.00059	76	21.053	0.00089	76	100.000	0.00156	20	2.843	0.00015	90	5.687	0.00028	83
0.2	0.3	0.6	0.000	0.00000	100	17.695	0.00101	53	5.505	0.00035	70	18.119	0.00113	53	100.000	0.00279	10	4.840	0.00074	90	4.507	0.00037	76
0.2	0.3	1	0.000	0.00000	100	16.779	0.00136	36	5.384	0.00077	70	10.172	0.00116	60	100.000	0.00539	10	3.132	0.00041	73	6.875	0.00115	63
0.2	0.6	0.3	0.000	0.00000	100	38.582	0.00101	53	7.716	0.00026	76	24.780	0.00093	66	100.000	0.00162	20	3.199	0.00024	93	17.315	0.00085	83
0.2	0.6	0.6	0.000	0.00000	100	25.733	0.00152	63	10.758	0.00089	80	14.367	0.00074	66	100.000	0.00326	23	8.220	0.00119	93	1.894	0.00018	90
0.2	0.6	1	0.000	0.00000	100	24.931	0.00350	53	5.581	0.00074	73	9.914	0.00122	66	100.000	0.00540	3	3.346	0.00057	76	4.163	0.00058	73
0.2	1	0.3	0.000	0.00000	100	38.881	0.00119	66	19.547	0.00066	76	17.139	0.00054	73	100.000	0.00134	23	3.470	0.00020	93	7.861	0.00031	86
0.2	1	0.6	0.000	0.00000	100	28.593	0.00136	53	9.234	0.00076	80	8.341	0.00050	66	100.000	0.00241	10	1.724	0.00022	93	6.987	0.00077	80
0.2	1	1	0.000	0.00000	100	16.553	0.00141	46	8.041	0.00106	70	5.702	0.00084	73	100.000	0.00483	6	2.729	0.00051	86	7.034	0.00127	80
0.8	0.3	0.3	0.000	0.00000	100	9.399	0.00047	86	0.707	0.00005	96	1.625	0.00007	90	100.000	0.00118	13	0.353	0.00003	96	3.180	0.00020	93
0.8	0.3	0.6	0.000	0.00000	100	4.981	0.00040	86	3.048	0.00030	90	5.428	0.00057	86	100.000	0.00245	13	1.375	0.00014	90	0.335	0.00005	96
0.8	0.3	1	0.000	0.00000	100	5.779	0.00079	73	4.620	0.00076	80	5.409	0.00065	73	100.000	0.00512	6	2.044	0.00045	83	2.254	0.00045	90
0.8	0.6	0.3	0.000	0.00000	100	4.960	0.00027	83	4.441	0.00027	86	8.535	0.00038	83	100.000	0.00130	6	0.000	0.00000	100	6.055	0.00028	83
0.8	0.6	0.6	0.000	0.00000	100	3.480	0.00028	83	1.501	0.00016	90	2.203	0.00019	83	100.000	0.00239	6	5.172	0.00072	90	0.926	0.00013	93
0.8	0.6	1	0.000	0.00000	100	11.687	0.00179	73	2.153	0.00050	90	1.245	0.00044	93	100.000	0.00482	6	3.632	0.00078	80	3.178	0.00049	76
0.8	1	0.3	0.000	0.00000	100	12.593	0.00047	86	6.667	0.00028	90	7.160	0.00024	83	100.000	0.00125	23	1.152	0.00004	90	5.350	0.00027	90
0.8	1	0.6	0.000	0.00000	100	5.013	0.00103	86	1.187	0.00023	93	5.211	0.00103	80	100.000	0.00345	3	2.177	0.00029	86	1.187	0.00020	86
0.8	1	1	0.000	0.00000	100	7.403	0.00111	76	6.966	0.00111	80	1.495	0.00028	86	100.000	0.00490	13	1.878	0.00029	86	1.240	0.00021	90
Overall Average			0.00000	0.00000	100	0.00056	0.00111	68	0.00021	0.00054	81	0.00029	0.00066	75	0.00370	0.00308	12	0.00011	0.00039	88	0.00016	0.00045	84