Minimizing Makespan for Machine Scheduling and Worker Assignment Problem in Identical Parallel Machine Models Using GA

Imran Ali Chaudhry, Sultan Mahmood and Riaz Ahmad

***Abstract*— Parallel machine scheduling, also known as parallel task scheduling, involves the assignment of multiple tasks onto the system architecture's processing components (a bank of machines in parallel). This paper presents a general purpose spreadsheet based genetic algorithm (GA) approach to minimize the makespan (total completion time) for a set of tasks for identical parallel machines and worker assignment to machines. The performance of the proposed approach is compared against two data sets of benchmark problems available on the internet. The proposed approach produces optimal solution for almost 95 percent of the problems demonstrating the effectiveness of the proposed approach.**

***Index Terms*—Genetic Algorithms, Identical Parallel Machines, Makespan, Scheduling, Worker Assignment.**

I. INTRODUCTION

Scheduling is a scientific domain concerning the allocation of limited tasks over time. The goal of scheduling is to maximize (or minimize) different criteria of a facility as makespan, occupation rate of a machine, total tardiness etc.

Parallel machine scheduling is important from both the theoretical and practical points of view. From the theoretical viewpoint, it is a generalization of the single machine scheduling problem. From the practical point of view it permits to take full advantage of the processing power provided by resources in parallel. Parallel machine scheduling comes down to assigning each operation to one of the machines and sequencing the operations assigned to the same machine. We may have identical, uniform or unrelated parallel machines. If the machines are identical, then the processing time of each job is the same on all machines. Uniform machines work at different speeds, i.e., the processing time of each job differs by a constant factor for the individual machines. If the machines are unrelated, then there is no relation between the processing times of the jobs and the machines.

In this paper we present a spreadsheet based GA approach to minimize the makespan for scheduling a set of tasks on identical parallel machines and worker assignment to the machines.

Manuscript received February 19, 2010.

I. A. Chaudhry (imran_chaudhryu@yahoo.com), S. Mahmood (sultanrandhawa@hotmail.com) and R. Ahmed (riazg19@gmail.com) are with Department of Industrial Engineering, College of Aeronautical Engineering, National University of Sciences & Technology, Risalpur, Pakistan. (imran_chaudhry@yahoo.com)

II. LITERATURE REVIEW

The parallel machine scheduling problems have been extensively studied in the literature [1]-[3]. Identical parallel machine scheduling problem with the makespan criterion is known to be NP-hard even for the case of two identical parallel machines [4].

Worker assignment scheduling problem has also been studied in the literature. In the classic parallel machine scheduling problem, no matter how many machines are involved, the number of workers at each machine may be ignored or assumed to be fixed and not taken into consideration. However, assigning more workers to work on the same job will decrease job completion time. Therefore, ignoring worker assignment decision may cause a managerial problem.

Hu [5]-[6] considered the parallel machines models with decisions for job scheduling and worker assignment to minimize total tardiness and total flow time respectively. A shortest processing time (SPT) heuristic and a largest marginal contribution (LMC) procedure were used to solve the job scheduling and worker assignment problems, respectively. The performance of these heuristics in [5]-[6] was further studied by Hu [7]-[8], who concluded that these heuristics generate the same results no matter what the value of W (number of workers).

Chaudhry and Drake [9] also considered the minimization of total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using GA. While Chaudhry [10] considered the minimization of total flow time for the worker assignment problem in identical parallel machine models using GA.

III. PROBLEM AND ASSUMPTIONS

In classical parallel machine scheduling problem, there are two essential issues to be dealt:

1. Partition jobs to machines
2. Sequence jobs for each machine

However, in the present research, the worker assignment scheduling problem needs to solve two sub-problems: how to assign jobs to machines and workers to machines? The objective is to minimize the makespan, which is defined as the total completion time of all the jobs. The performance of GA is compared with SPT/L(east)PA and L(argest)PA heuristics heuristic approach by Hu [11].

We assume that all machines are identical such that the processing time of a job is independent of machine.

Deterministic processing times and due dates are assumed. Machine setup times are included in the processing time. The only difference between worker assignment scheduling problem and the classic scheduling problem is that the former has an additional constraint of worker assignment also. The processing times of the jobs are therefore dependent on the number of workers assigned to work on a particular machine.

The following notation is used to define the problem.

A_i, B_i, E_i — integers, follow uniform distribution, such that $0 \le A_i < 10$, $0 \le B_i \le 50$, $0 \le E_i \le 10$ (1st data set); $0 \le A_i < 10$, $0 \le B_i \le 800$, $0 \le E_i \le 10$ (2nd data set)

n — number of jobs

m — the number of parallel machine in the shop.

p_i — the process time of job i.

W — number of workers in the shop.

Wj — number of workers assigned to machine m_j.

We have to assign n jobs to m parallel machines taking into account the following characteristics:

1) Each job has only one operation.
2) The jobs have different processing times which depend on the number of workers assigned to that machine.
3) No job pre-emption / splitting is allowed;
4) Each job has its own due date;
5) The selected objective is to minimize the total tardiness.
6) Any machine can process any job.
7) Machine setup times are negligible.
8) No machine may process more than one job at a time.
9) Transportation time between the machines is negligible.
10) Number of jobs and machines are fixed.
11) There is a group of W workers that have the same abilities to perform the duties assigned to them.
12) All m machines, n jobs and W workers are available at time zero.
13) Assume processing time function has a simplified form of $p_i(W_j)=A_i+B_i/(E_i \times W_j)$ where $p_i(W_j)$ is the processing time of job I that is processed on machine j in which the number of W_j workers are assigned on it, A_i is a fixed constant and not affected by the number of workers, $B_i/(E_i \times W_j)$ is the variable part and affected by the number of workers.
14) The number of workers assigned to each machine needs to be decided before any job can be processed and they will not be re-assigned until all the jobs have been completed.

The simulation experiments have been carried out for 12 jobs to be scheduled on three parallel machines with 10 workers.

IV. GENETIC ALGORITHMS

GAs is one of problem solving systems based on the principles of evolution and hereditary, each system start with an initial set of random solutions and use a process similar to biological evolution to improve upon them that encourage the survival of the fittest. The best overall solution becomes the candidate solution to the problem. A detailed introduction to Gas is given in Goldberg [12]. The earliest application of GA has been reported by Davis [13]. For a recent review of GA application in scheduling is given in Chaudhry and Drake [9] and Chaudhry [10].

In this study, the tool use for carrying out the GAs is a commercial software package called EVOLVER™ [14], which functions as an add-in to Microsoft Excel™. The objective function, variables (adjustable cells), and the constraints are readily specified by highlighting the corresponding spreadsheet cells. The scheduler/evaluator portion of the model is constructed by using the spreadsheet's built in function. The schematic in Fig. 1 illustrates the integration of the GA with the spreadsheet.

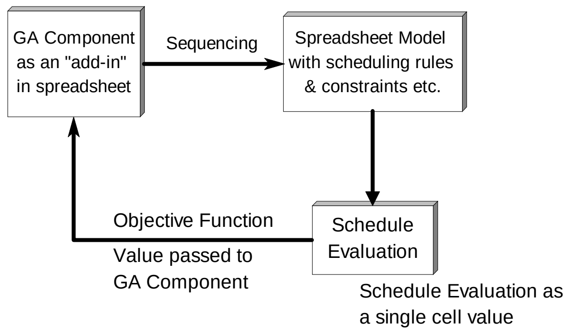

Fig. 1: Integration of GA and spreadsheet

The advantage of the proposed method is that the program runs in the background freeing the user to work in the foreground. When the program finds the best result it notifies the user and places the values into the spreadsheet for analysis. This is an excellent design strategy given the importance of interfacing with spreadsheet in business.

A. *Chromosome Representation*

The chromosome representation for the machine scheduling and worker assignment problem is shown in the Fig. 2.

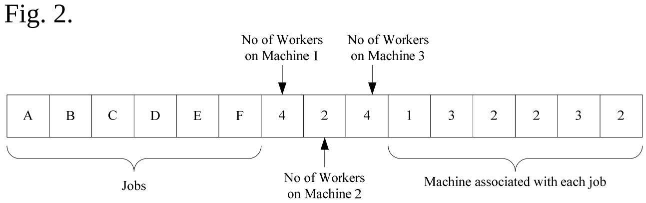

Fig. 2: Chromosome Representation

For determining the ordering of the jobs i.e., for the first six genes we needed to handle the permutation representation. The next three genes represent the number of workers assigned to each machine stating that four workers are assigned to machine 1, two workers to machine 2 and four workers to machine 3. While the machine assignment block can be read as the assignment of jobs to each of the six machines, whereby the assignment of the jobs to machines would be as follows: Job A to be processed on machine 1; job B on machine 3, job C on machine 2, job D on machine 2, job E on machine 3 and job F on machine 2 respectively.

Each of the block in Fig 2 is calculated at a different location in the spreadsheet which is in turn linked to the calculation of the objective function i.e., the makespan.

B. *Reproduction/Selection*

Evolver uses a steady state approach. This means that only one organism rather than an entire population is replaced at a time. The number of generations can be found by dividing the

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

trials by the size of the population. As far as parent selection is concerned, in Evolver, parents are chosen using a rank-based mechanism. This procedure begins by rank ordering the population by fitness. Next an assignment function gives each individual a probability of inclusion. The assignment function can be linear or nonlinear. A roulette wheel is then built with the slots determined by the assignment function. The next generation of an *n*-sized population is built by giving the wheel *n* spins. This procedure guides selection towards the better performing members of the population but does not force any particular individual into the next generation. Fig 3 describes the steady state algorithm.

Repeat
    Create n children through reproduction
    Evaluate and insert the children into the population
    Delete the n members of the population that are least fit
Until stopping criteria reached

Fig. 3: Steady State Algorithm

C. *Crossover Operator*

As for the first six genes of the chromosome, we needed to handle permutation representation; the "Order Solving Method" of Evolver was used. This method applies order crossover operator [13]. This selects items randomly from one parent, finds their place in the other parent, and copies the remaining items into the second parent in the same order as they appear in the first parent. This preserves some of the sub-orderings in the original parents while creating some new sub-orderings. Fig 4 shows the crossover operator as described above.

Position:	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***
Parent 1 (P1):	1	2	3	4	5	6	7	8	9
Binary Template:	*0*	*1*	*1*	*0*	*1*	*1*	*0*	*0*	*1*
Parent 2 (P2):	4	9	5	8	3	6	7	1	2
Offspring (O):	4	2	3	8	5	6	7	1	9

Fig. 4 Order Crossover Operator

For the number of workers and machine assignment the "Recipe Solving Method" of Evolver was used. The "Recipe Solving Method" implements uniform crossover [13]. This means that instead of chopping the list of variables in a given scenario at some point and dealing with each of the two blocks (called "single-point" or "double-point" crossover), two groups are formed by randomly selecting items to be in one group or another. Traditional x-point crossovers may bias the search with the irrelevant position of the variables, whereas the uniform crossover method is considered better at preserving schema, and can generate any schema from the two parents. For worker assignment (genes 7–9 in Fig. 2) and machine assignment (genes 10–15 in Fig. 2), we have a set of variables that are to be adjusted and can be varied independently of one other. In the spreadsheet model presented in this paper, the constraint placed on the workers and machines is to set the range that the variable must fall between. Similarly, as there are ten workers available in the shop, another constraint that ensures that only valid solutions are retained in the population is that the sum of values generated for genes 7–9 should be equal to 10. This ensures that no invalid solution is retained in the population pool. Fig 5 shows a typical uniform crossover operator.

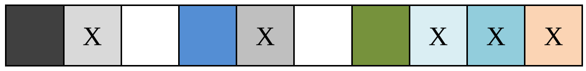

Fig. 5 Uniform crossover – a given % of the organism is randomly selected

D. *Mutation Operator*

To preserve all the original values, the "Order Solving Method" performs mutation by swapping the positions of some variables in the organism. The number of swaps performed is increased or decreased proportionately to the increase and decrease of the mutation rate setting (from 0 to 1).

The "Recipe Solving Method" performs mutation by looking at each variable individually. A random number between 0 and 1 is generated for each of the variables in the organism, and if a variable gets a number that is less than or equal to the mutation rate (for example, 0.06), then that variable is mutated. The amount and nature of the mutation is automatically determined by a proprietary algorithm. Mutating a variable involves replacing it with a randomly generated value (within its valid min-max range).

V. EXPERIMENTAL RESULTS

A. *Implementation Details*

The job order (first six genes) in Fig. 2 is a permutation of a list of jobs where we are trying to find the best way to arrange a set of given jobs. This permutation is independent of the number of workers on each machine and assignment of job to a particular machine. However, the objective function is calculated keeping in view all the constraints which are discussed in the next paragraph.

For number of workers on each machine and the machine corresponding to each job, i.e., for genes 7-9 and 10-15 (Fig. 2), random integer numbers are generated by the GA subject to the constraints that have been defined in the initial setup of the GA. Constraints are basically the conditions that must be met for a solution to be valid. The constraint imposed on the number of workers on each machine is that the sum of all workers assigned to the three machines should always be 10, which is the total number of workers available in the shop. The range for random integer is from 1 to 10. Therefore, only those solutions are kept in the population where the sum of all workers is 10. Hence, only integer values between 1 and 10 (both limits inclusive) are generated, while the constraint for machine corresponding to each job is that an integer number is selected from among 1, 2, and 3 (which are machine numbers).

B. *Computational Analysis*

In order to check the effectiveness of the proposed spreadsheet based GA approach two data set of 100 problems each [11] were used. The problems in both the data sets have 12 jobs to be scheduled on three identical parallel machines where the number of workers available is equal to 10. The only difference among the two data sets in terms of the value of the variable B_i, for first data set it is $0 \le B_i \le 50$ and for second data set it is $0 \le B_i \le 800$. The problems have been

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

simulated on a P1V 1.7 GHz computer having 512 MB RAM. By conducting repeated tests, we found that the best values to be set for the number of population, the crossover rate, and the mutation rate are 60, 0.65, and 0.09, respectively. Therefore, for each of the run, same set of parameter setting as described previously have been used, which correspond to 3 min 20 s on a PIV 1.7 GHz computer having 512 MB RAM.

Table 1 and Table 2 give a summary of the results for the first and second data sets respectively. The summary shows the performance of proposed GA approach with the SPT/L(east)PA and L(argest)PA heuristic proposed by Hu [11].

Table 1: Summary of results for the two approaches - 1st data set

		GA	SPT/L(east)PA and L(argest)PA heuristic [11]
# of problems simulated		100	100
# of optimal solution obtained		95	22
Max percentage error		7.87%	65.47%
Avg Time to find best solution		70 secs	-
Comparison of GA with other methods	Same	-	22
	Better	-	77
	Worse	-	1

Table 2: Summary of results for the two approaches - 2nd data set

		GA	SPT/L(east)PA and L(argest)PA heuristic [11]
# of problems simulated		100	100
# of optimal solution obtained		96	18
Max percentage error		8.94%	38.31%
Avg Time to find best solution		86 secs	-
Comparison of GA with other methods	Same	-	27
	Better	-	71
	Worse	-	2

GA produced superior solution as compared to the SPT/L(east)PA and L(argest)PA heuristic. For the first data set GA found 95 optimal solutions as compared to the heuristic which found optimal solution for 22 problems. While for second data set these values were 96 and 27 respectively. The average time for the GA for first and second data set was to find the best solution was 70 secs and 86 secs respectively. As compared to the optimal solution, for the first data set the maximum percentage error was 7.87% and 65.47% for the GA and SPT/L(east)PA and L(argest)PA heuristic [11] respectively. While for the second data set these maximum percentage errors were 8.94% and 38.31% respectively.

As compared to SPT/L(east)PA and L(argest)PA heuristic, for the first data set GA produced same solution for 22 problems and better for 77 problems. For the second data set these were 27 and 71 respectively. While for only one problem in first data set and two problems in second data set GA produced worse results as compared to SPT/L(east)PA and L(argest)PA heuristic. Detailed comparative results of GA, SPT/L(east)PA and L(argest)PA heuristic and Optimal solution are given in Table 3 and Table 4.

VI. Conclusion

This paper presented a spreadsheet based general purpose GA solution methodology for the scheduling of set of job on parallel machines and worker assignment to machines. The spreadsheet GA implementation has been found to be easy to implement catering for the peculiarities of any environment. Moreover, the spreadsheet environment makes it very suitable to carry out what if analysis. The results in Table 1 and 2 clearly show the superiority of proposed GA approach as compared to an earlier study by Hu [11]. The spreadsheet model can be easily customized to include additional jobs, machines or workers without actually changing the logic of the GA routine thus making it a general purpose scheduling approach.

The key advantage of GAs portrayed here is that they provide a general purpose solution to the scheduling problem which is not problem-specific, with the peculiarities of any particular scenario being accounted for in fitness function without disturbing the logic of the standard optimization routine. The GA can be combined with a rule set to eliminate undesirable schedules by capturing the expertise of the human scheduler.

References

[1] B. Chen, C. N. Potts, G. J. Woeginger, “A review of machine scheduling: Complexity, algorithms and approximability”, in: D. Z. Du, P. M. Pardalos (Eds.), *Handbook of Combinatorial Optimization*, Kluwer Academic Publishers, Dordrecht, 1998, pp. 21–169.
[2] T. C. E. Cheng, C. C. S. Sin, “A state-of-the-art review of parallel-machine scheduling research”, *European Journal of Operational Research*, vol. 47, 1990, pp. 271-292.
[3] E. Mokotoff, “Parallel machine scheduling problems: A survey”, *Asia-Pacific Journal of Operational Research*, vol. 18, 2001, pp. 193-242.
[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.
[5] P-C. Hu, “Minimizing total tardiness for the worker assignment scheduling problem in identical parallel-machine models”, *International Journal of Advanced Manufacturing Technology*, vol. 23(5–6), 2004, pp. 383-388.
[6] P-C. Hu, “Minimizing total flow time for the worker assignment scheduling problem in the identical parallel-machine models”, *International Journal of Advanced Manufacturing Technology*, vol. 25(9–10), 2005, pp. 1046-1052.
[7] P-C. Hu, “Further study of minimizing total tardiness for the worker assignment scheduling problem in the identical parallel machine models”, *International Journal of Advanced Manufacturing Technology*, vol. 29, 2006, pp. 165-169.
[8] P-C. Hu, “Further study of minimizing total flowtime for the worker assignment scheduling problem in the identical parallel machine models” *International Journal of Advanced Manufacturing Technology*, vol. 29(7–8), 2006, pp. 753-757.
[9] I. A. Chaudhry and P. R. Drake, “Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms”, *International Journal of Advanced Manufacturing Technology*, vol. 42, 2008, pp. 581-594.
[10] I. A. Chaudhry, “Minimizing flow time for the worker assignment problem in identical parallel machine models using GA”, *International Journal of Advanced Manufacturing Technology*, 2009, DOI 10.1007/s00170-009-2323-1.
[11] P-C. Hu. Online research data set available at http://sparc.nfu.edu.tw/~pchu/research_data.htm.
[12] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, 1989, Addison-Wesley, Boston.
[13] L. Davis, *Handbook of Genetic Algorithms*, 1991, New York, Van Nostrand Reinhold
[14] Palisade Corporation, Evolver: the genetic algorithm super solver. 1998, New York, USA

Table 3. Comparative results of GA, Hu [11] and the Optimal Solution for first data set

Prob #	GA Solution	% Deviation		Time (secs)	Prob #	GA Solution	% Deviation		Time (secs)
		Hu	Optimal				Hu	Optimal	
1	15.4028	-6.1608	0	125	51	16.8833	-17.6873	0	70
2	12.0188	0	0	71	52	12.6659	-10.8187	0	93
3	16.9714	-4.8331	0	81	53	14.6190	0	0	23
4	19.5708	0	0	45	54	15.6667	0	0	27
5	18.6333	0	1.8215	90	55	14.1892	-5.8185	0	88
6	17.7444	7.4335	7.8690	103	56	15.9143	-13.9510	0	89
7	16.8576	-1.4813	0	5	57	17.7500	-3.1818	0	7
8	21.5556	-8.5983	0	118	58	16.1000	-13.5957	0	64
9	11.9360	-15.2475	0	65	59	15.3684	0	0	101
10	22.7583	-15.5087	0	109	60	11.9250	0	0	112
11	15.5556	-7.8189	0	154	61	15.5872	-9.3473	0	129
12	18.4583	-6.5401	0	67	62	18.0635	-4.1583	0	190
13	15.2361	0	0	75	63	15.8411	-39.5667	0	99
14	14.6670	-5.1618	0	29	64	18.7625	0	0	10
15	14.1222	-12.5017	0	67	65	17.2250	-3.8605	0	6
16	16.3125	-13.8298	0	24	66	16.1000	-13.7981	0	19
17	11.4555	-11.9024	0	33	67	16.6898	-9.1613	0	26
18	14.2000	-11.5737	0	166	68	21.4486	-7.4541	0	185
19	12.0696	-13.2962	0	66	69	16.5736	-1.2986	0	41
20	15.5417	-15.1887	0	101	70	13.9278	-14.4310	0	66
21	13.0778	-15.3307	0	36	71	15.2231	-1.4330	0	4
22	12.3624	-9.6167	0	38	72	18.3542	-2.2559	0	31
23	17.6806	-6.7991	2.9340	9	73	14.3050	0	0	60
24	14.5986	-9.3254	0	49	74	10.4955	-12.0968	0	30
25	19.1250	-21.9830	0	21	75	15.1569	-2.1212	0	138
26	12.7429	0	0	40	76	11.7333	-21.5994	0	60
27	11.4272	0	0	35	77	17.4120	0	0	7
28	15.3810	-10.9211	0	6	78	17.2738	0	0	191
29	20.2965	-3.0615	0	167	79	15.3125	-5.7692	0	68
30	17.5556	-4.7093	0	64	80	11.4583	0	0	96
31	10.3304	-19.6568	0	219	81	15.2297	-7.3559	0	9
32	24.8259	-5.4716	0	25	82	18.7500	-2.5974	0	77
33	18.6190	-6.1224	0	108	83	18.4005	0	0	82
34	18.1389	-11.8758	0	18	84	14.0000	-13.4962	0	62
35	16.1647	-3.2193	0	38	85	13.2738	-2.1758	0	84
36	14.0000	-9.2012	0	52	86	13.1414	-12.1818	0	20
37	18.1690	-10.3828	0	147	87	15.1111	-3.3502	0	17
38	20.5635	-12.1453	0	9	88	16.2143	-7.4087	0	106
39	17.3006	-5.9791	0	41	89	16.2083	-12.4550	0	8
40	16.0840	-8.0393	0	186	90	13.6000	-2.9534	0	19
41	14.9256	-24.7374	0	43	91	16.6833	-11.1125	0	197
42	11.0000	0	0	76	92	12.1528	0	0	3
43	16.9048	-5.9914	0	64	93	14.6984	-9.5631	0	13
44	13.8086	0	0	44	94	15.3643	-5.4505	1.1512	143
45	14.6594	-14.3759	0	132	95	17.4369	0	0	94
46	12.3583	-8.0025	0	53	96	12.1131	-16.3653	0	165
47	14.6741	0	0	43	97	15.9310	-15.0727	0	43
48	16.8438	-5.5049	0	21	98	17.5094	-5.5435	0	126
49	13.7434	-10.5729	0	92	99	13.7000	-7.7556	0	23
50	14.1447	-8.2035	0.6047	33	100	10.9602	0	0	75

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table 4. Comparative results of GA, Hu [11] and the Optimal Solution for second data set

Prob #	GA Solution	% Deviation Hu	% Deviation Optimal	Time (secs)	Prob #	GA Solution	% Deviation Hu	% Deviation Optimal	Time (secs)
1	66.5733	0	0	26	51	58.9960	-4.0622	0	194
2	53.1369	-13.0724	0	102	52	81.8125	-27.6996	0	93
3	70.2032	0	0	11	53	62.7251	0	0	3
4	98.4630	-3.1410	0	157	54	69.3333	-1.7794	0	44
5	95.5000	-2.4728	0	211	55	64.2640	-3.3163	0	142
6	67.3482	-9.7483	0	3	56	89.2130	0	0	228
7	86.0556	-7.1343	0	48	57	92.1333	0	0	4
8	88.5806	-3.7517	0	194	58	76.9896	-4.4597	0	63
9	64.3122	-5.2478	0	107	59	76.8333	-12.0259	0	31
10	80.7827	-8.2666	0	35	60	60.9722	-14.4983	0	166
11	76.3681	-7.3765	0	117	61	57.3472	-11.0896	0	179
12	60.4294	0	0	96	62	81.6071	-9.7670	0	1
13	71.6310	-9.6094	0	91	63	81.3820	-3.4479	0	16
14	60.5625	0	0	123	64	68.1190	1.7787	8.9421	4
15	69.3472	0	0	98	65	84.3519	-6.4605	0	17
16	72.7496	-8.5996	0	16	66	92.4063	-19.4132	0	15
17	50.2143	-2.9490	0	42	67	103.8519	-13.0178	0	117
18	82.1771	-13.5989	0	1	68	83.5978	-1.1814	0	16
19	74.2363	0	0	18	69	71.3333	-16.1880	0	7
20	70.3631	-7.6383	0	172	70	79.5132	-2.3046	0	186
21	84.1667	-1.7510	0	3	71	76.7500	-1.6676	0	130
22	54.3905	0	0	200	72	88.3750	-10.7113	0	105
23	65.4854	-16.5179	0	122	73	92.6833	0	0	118
24	70.3228	0	0	3	74	84.9747	0	0	171
25	68.6111	0	0	2	75	85.6762	-7.2159	0	9
26	53.1111	-10.6124	0	59	76	53.2900	0	0	198
27	57.2460	0	0	172	77	70.7817	0	0	144
28	69.0313	-0.8272	0	131	78	90.8929	-8.4203	0	10
29	94.2116	-13.1635	0	2	79	79.7500	-0.8290	0	150
30	85.1333	-4.3983	1.3010	5	80	65.7223	-4.2782	0	44
31	67.4450	-5.2107	0	94	81	66.4901	-12.3850	0	58
32	114.7063	-4.7704	0.9623	9	82	88.3889	-6.4903	0	22
33	67.7917	-9.6334	0	5	83	115.7188	-8.7571	0	398
34	72.6310	-11.0387	0	174	84	65.4980	0	0	145
35	88.9722	0	0	1	85	87.4861	0	0	297
36	60.5000	-5.2219	0	35	86	62.1778	-6.1299	0	109
37	82.1600	-4.4327	0	68	87	83.1667	-12.0550	0	3
38	80.3375	-6.2968	0	286	88	77.3433	-12.7462	0	5
39	62.5693	0	0	54	89	112.7986	0	0	76
40	72.5040	-14.2348	0	3	90	70.3979	0	0	190
41	68.3968	-8.0324	0	29	91	89.6296	-6.2561	0	86
42	51.5878	0	0	4	92	72.1116	-4.0416	0	114
43	64.4464	-0.4711	0	246	93	85.1333	3.3673	4.0750	40
44	51.1500	-15.0755	0	181	94	70.8854	-3.2831	0	215
45	62.4352	-5.9301	0	75	95	80.4802	-6.2887	0	7
46	62.1250	-4.9749	0	2	96	70.0463	-5.9871	0	121
47	72.8630	0	0	118	97	62.6976	-8.4839	0	92
48	77.8750	0	0	5	98	70.6032	-10.7746	0	2
49	65.8417	0	0	2	99	61.4889	-2.2212	0	191
50	56.6508	0	0	2	100	65.9468	-21.6430	0	33

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)