Multiobjective Optimization of the Train Staff Planning Problem using NSGA-II

Simon Girardin
School of Business
FHNW University of Applied Sciences and Arts Northwestern Switzerland
Olten, Switzerland
simon.girardin@students.fhnw.ch

Fabian Baumann
School of Business
FHNW University of Applied Sciences and Arts Northwestern Switzerland
Olten, Switzerland
fabian.baumann@students.fhnw.ch

***Abstract*—The optimization of assigning train staff to scheduled train services is called the train staff planning problem. A part of this is rostering with the aim to create a duty timetable under the consideration of different constraints, preferences etc. In this paper, we propose and analyze the application of the nondominated sorting genetic algorithm II (NSGA-II) Algorithm for multiobjective optimization of the rostering.**

Keywords—Train staff planning, railway crew, crew planning, crew rostering, crew scheduling, NSGA-II, Multiobjective optimization

I. Introduction

The train staff planning problem (TSPP) is an assignment problem where human resources like train drivers and conductors need to be assigned to a defined set of tasks hereby to scheduled train services [1]. Usually, train staff planning is the task to create a duty timetable for a planned train timetable. The structured approach often is to plan first the certain lines of a network and make a timetable with arrival and departure times of each train. Afterword, the so-called train platforming is done to make sure the stations can handle the number of trains at certain times. Then with the rolling stock scheduling, the specific trains are assigned. Next, comes the task to plan the crew members. This is usually divided into two sub-phases. The staff scheduling problem and staff rostering problem. In the scheduling phase, the train services are assigned to generic staff whereas the rostering phase schedules the individual and specific staff. Each of these phases are very complex because of a variety of constraints [2]. In this paper, we concentrate to solve the crew rostering problem. So we assume the previous steps have already been executed including the staff scheduling. Therefore services are created and the specific staff members (employees) need to be assigned to this services. The problem model with their objective function and constraints can differ very much, depending on the objective and the specific case. For example the size of the problem from a small scale regional or urban train network to larger long-distance ones [3]. The solving of this problem has many similarities to other rostering problems, especially those in the transportation industry.

In the literature, on the TSPP one can find several objectives that have been optimized. In [1] the penalization of hard and soft constraints was minimized in a single as well as multiobjective approach. Another objective can be seen in [4], where the goal is the minimization of the total cost of assigned train drivers to duties during the planning period. Also, a large variety of constraints have been taken into consideration. For example, constraints which are depending on the availability of employees like the non-availability of drivers due to vacation [1], [5]. Legal aspects such as rest days, rest time between shifts, or total week working time have been considered [4]–[6]. Soft constraints can be preferences by the staff or the transportation company or constraints that are allowed to be violated. Examples are favorability of specific assignments for drivers to specific rolling stock [1]. Approaches for the optimization model can be from integer programming methods [5] to different heuristic or metaheuristic algorithms [7]. The problem can be solved with a single-objective as well as a multiobjective approach. The use of the multiobjective approach has several advantages including allowing the decisionmaker to consider his personal opinions by showing him different possible solutions [1]. Several different metaheuristic algorithms allow such an approach. For example, multiobjective simulated annealing was applied in [8], but with moderate success with regard to the computing time.

Our work contributes primarily by proposing the use of the nondominated sorting genetic algorithm II (NSGA-II) to solve the train staff planning problem (TSPP). This approach was successfully done by [9] for airline crew scheduling problems and by [10] for home care staff planning but it has never been done for the TSPP. The NSGA-II Algorithm has in both mentioned studies performed better than other Multiobjective Evolutionary Algorithms (MOEAs) like the Strength Pareto Evolutionary Algorithm (SPEA2) or the Non-dominated Ranking Genetic Algorithm (NRGA) [10], [11].

The structure of this paper is organized as follows. The solved problem is described in chapter II, followed by the optimization model with the objective functions as well as constraints and preferences in chapter III. After that, a description of the NSGA-II algorithm and its implementation on the described problem is given in Chapter IV. The results are stated in chapter V. Finally, a conclusion with the critical acclaim of this paper in VI.

II. Description of the Optimization Problem (OP)

Because there is no public available sample data that can be used for testing this approach we have created a simple case by our self. In our theoretical case, we assume that the crew scheduling has already been completed and that the individual train drivers must now be assigned to a set of services. Each train driver has his place of residence as a start point for his work. Each train driver has their individual preferences on train-type and shifts. In total there is a set of ten train drivers in this case. The set of services consists of a total of 63 services which are spread over one week. This results in nine services

per day which in turn are distributed over three shifts. In addition to each shift, a specific train type is assigned. We made sure that the same train is used for a new shift from the location where it was parked during the last shift. A service can start and end at four different locations A, B, C, and D of the train network as shown in fig. 1.

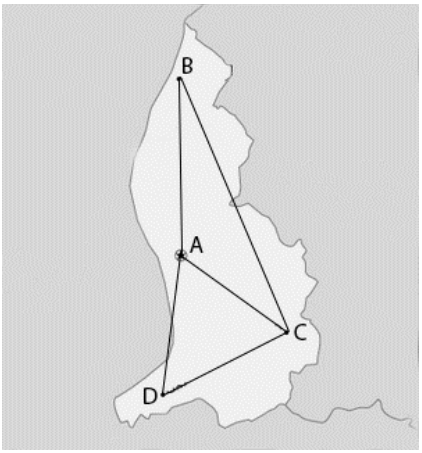

Fig. 1. Line network map of the theoretical case

As crew costs are the largest cost factor in the rail passenger transport we, therefore, try to reduce this cost with an optimized rostering [12]. The train driver costs as soon he has left his place of residence till he returns at home. A minimized approach route to the first trip and teaching methods should be minimized to save costs. With the second goal, we try to take the preferences of the staff into account. Consistently having different shifts is uncomfortable and a nuisance. A well-designed work schedule becomes often a central issue for the staff satisfaction and therefore in the negotiation of collective agreements [5].

III. Identifying the Problem Model (PM)

A. Objective Functions

Multiobjective optimization that we applied, consists of two different objective functions. With the first objective, we do a minimization of penalties for violation of preferences (p) (soft constraints) and hard constraints (m). For each solution, we calculate the sum of the penalty points for preferences and constraints. The penalizing factor is described in the next section of this paper.

$$f = \sum_{N}\sum_{p} u_p + \sum_{N}\sum_{m} w_m \tag{1}$$

f = objective function
i = number of services to be planned
u_p = penalizing factor for preference p
w_m = penalizing factor for constraints m
N = set of services

The second objective is to reduce the total labour costs. Here we calculate the costs for the journey from the employee resident location (er) to the designated start location (ss) of his assigned service and the cost for the return journey from the end location (se) of the service to his resident location. The mathematical formulation of this can be seen in equation (3). In our case, the costs for the journey are read from a table. Because the time during the shift does not vary, no matter which employee is assigned, we do not count this. We also assume that the cost for each employee is the same. The objective function can be seen in (2) whereas in (3) a cost of one certain service is formulated.

$$f = \sum_{s \in N} c_s \tag{2}$$

subject to:

$$c_s = c_{ss}^{er} + c_{er}^{se} \tag{3}$$

s = service
c_s = cost per service
er = employee resident point
ss = service start location
se = service end location
c = costs

B. Hard Constraints

Because we have three shifts a day and there are services seven days a week in our case, we have to consider some hard constraints when calculating the solutions. This to avoid that employees are assigned to services they cannot attend due to legal reasons such as rest time and maximum work hours per day. The hard constraints need to be fulfilled. Otherwise, they will be seen as not feasible solutions.

1) A service must have an existing employee assigned

2) An employee cannot have 2 shifts on the same day

3) An employee cannot have 2 shifts in a row

If one of these hard constraints is not fulfilled the fitness of the solution will be penalized with -1000 points.

C. Soft Constraints

To respect the preferences of each employee we consider following soft constraints that are penalized if these are not observed.

1) Each employee can choose his shift preference or he can also deselect the preference if he doesn't care. If this constraint is not met, the fitness will be penalized with -3 points.

2) Each employee can also choose his preferred train type. This constraint will be penalized with -2 points if it is violated in a solution.

3) In addition, the goal is to assign a train driver to the same shift during the week. If this can not be achieved a penalty of -5 points is given.

For our case, we made a table with randomly selected preferences for each employee. The preference categories are favorite train type, favorite shift, and whether they prefer the same shift in one week or not with true or false.

IV. Optimization Method (OM) & Algorithm

As mentioned in the introduction we have chosen the NSGA-II algorithm as the optimization method.

A. The nondominated sorting genetic algorithm II (NSGA-II)

The by [11] proposed NSGA-II approach has been chosen as an optimization method for the above described problem because it is one of the most popular algorithms for

multiobjective optimization. The concept of the NSGA-II mainly builds on the non-dominated sorting and crowding distance assignment. Furthermore, it gives out a parento-optimal front. The algorithm works as follows: First a random parent population (P_0) of the size L is created. Then after calculating the fittnes for each solution, a offspring (child) population (Q_0) of the same size L is created with binary tournament selection and mutation. Both populations combined (R_t) are then ranked and sorted with nondomination rank to F_1 the best one, F_2 the second best etc. To then create a new population (P_t+1) crowding distance soriting is applied. With crossover and mutation the offset population is created again before all starts over by combining both populations.

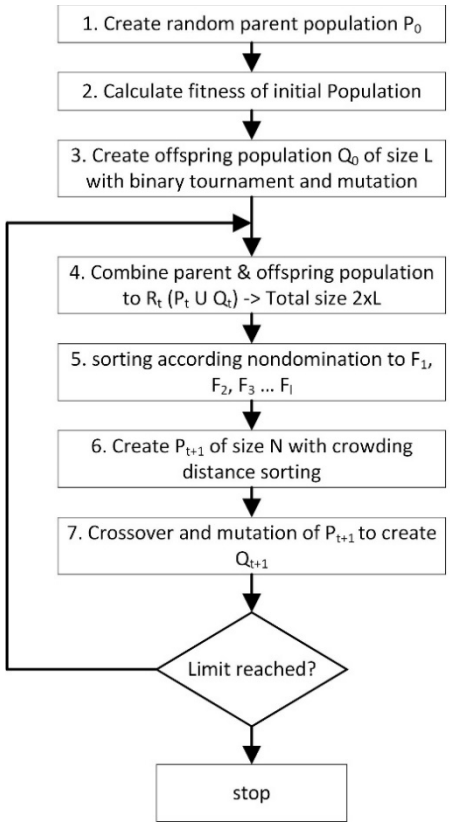

Fig. 2. Flow chart of NSGA-II

In the implemented algorithm, the result of the fitness function is the aggregation of the results of the objective functions. To calculate the fitness of a chromosome, it has to be handeld by all objective functions that give each a score. The score is calculated in accordance of the different constraints that are respected or not. This score gives the coordinate on the axis to whom the objective function dimension is linked to on the chart.

B. Chromosome representation

We decided to use a binary coding scheme. One chromosome represents the total rostering plan. Each gene in the chromosome represents an employee-ID. We have a total of ten employees, so one gene consists of four alleles (bits). The locus (position) of a gene in the chromosome represents a specific service. Thus, every service is present in the chromosome and none is left without allocation. The disadvantage is, however, that an employee ID can be chosen which does not exist or which conflicts with the hard constraints. We try to take both problems into account by punishing them severely if they occur. See also section hard constraints in this paper. However, for the initialization, when creating the first parent population P_0, we have made sure that only valid Employee-ID are taken. With a gene length of four and a total of 63 services, the length of a chromosome in our case is 252 bits.

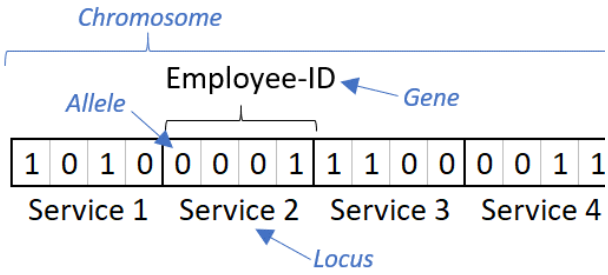

Fig. 3. Coding scheme of the chromosome

C. Operators of Algorithm

The NSGA-II Algorithm allows us to configure some operators. In this study uniform crossover by default is applied for all calculations. This means that for the crossover the bits for a new child will get randomly from the parent chromosomes. Also, we did not change the mutation schemes from the original paper [11] and have chosen the single-point mutation here. This causes random bits of a chromosome to change. But the program checks if the new employee-ID gene that is generated through mutation, is valid. If not, the mutation will continue till a valid Employee-ID is created. To check the validity of new genes, we had to group the new alleles in a gene sized list. Then the content of this list was verified against the Employe-ID list generated at the start. See Fig. 4 which describes the steps of this process.

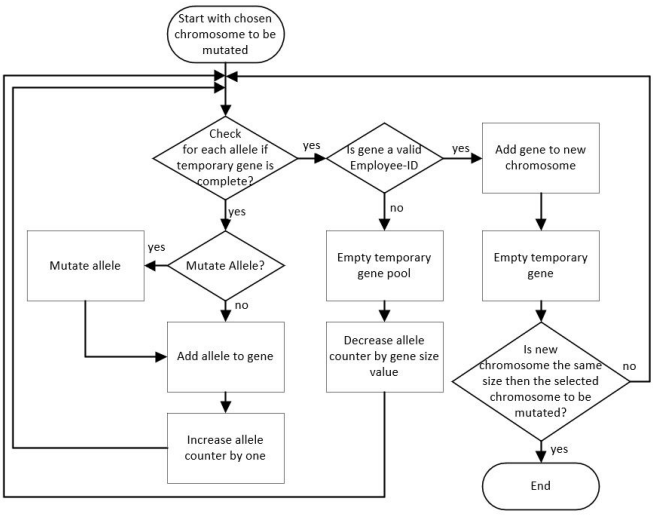

Fig. 4. Flow chart of the mutation operator in our program

Several variants of operators can be set and were tested in this work and are discussed in the results section. The program allowed us to change the number of generations which means the number of iteration loops of the NSGA-II. The population size which is the initial population as well as the ones in each generation. We also could change the probability of crossover and probability of mutation.

D. Software implementation

We made the implementation using a Java program and executed it in IntelliJ IDEA. As a basis for the program, we

used the code of [13] that translated the C code to Java from the original paper [11] and extended and adapted it for our problem. As the code from this source was for a simple mathematical problem we changed the way how the chromosomes are coded as described before in this paper. Also, we integrated the possibility of adding constraints and changed the way how the fitness is calculated according to our requirements like described earlier.

The main challenge was the representation of the genes. In the original paper and program, only chromosomes and alleles are used. The notion of genes didn't exist, thus we needed to adapt the codebase at multiple places since this is a central part of our problem model. The software can be accessed on Github repository[1].

V. Results and Discussion

To find a good configuration of the NSGA-II algorithm we have done several test runs. The goal was to achieve good fitness results on both axes and to generate a meaningful Pareto front. We always kept the starting position the same. In each run, we only changed one parameter to see the effects.

We started by slightly increasing the population size. During the experiment, we found that the best results were achieved with a population size of 300. With this value fixed the number of generations was increased step-by-step. We have also evaluated that the best results are achieved from about 200 generations on. For the probability of mutation and crossover we had assumed average values since the beginning of the experiment. Surprisingly, the best results were achieved when we selected a very low mutation probability. The crossover probability was best at a value of 0.4f. A table with all tests and their different parameters including their results can be found in the appendix. Figures 4 and 5 shows the solutions graphically represented the test run with ten employees and the best setting of the algorithm.

Because the original program maximizes the first objective on the y-axes, but we want to minimize both, penalties and costs, we have reversed the scale of the costs by multiplying it with -1. This means that if you read the diagram, the upper right-hand corner is the optimum for our case. So the least penalized solution at the lowest cost.

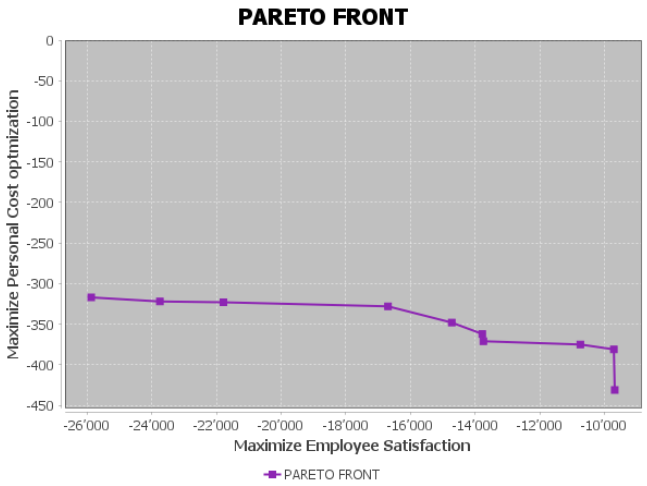

Fig. 5. Pareto front of the best configuration found with 10 employees

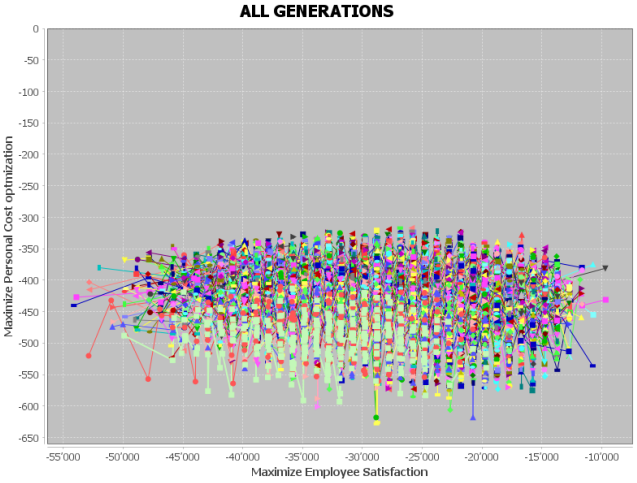

Fig. 6. All Solutions of the best configuration with 10 employees

Despite our improvements in the parameters, relatively high penalty values have still been occured in the preferences. The reason for this is that some hard constraints per solution were not considered. We have therefore adjusted the size of the train crew to 15 and 20 train drivers. The adjustment to 20 persons had the consequence that the number of alleles of the gene increased by one. The size of a crew with 15 employees has the adventage that only feasible employee-ID could be created as 15 is the maximum of a four bit gene. But as in real cases the number of employees are not selected according to the gene size this is only a theoretical test. With this increase, we could achieve significantly better solutions in terms of penalizing preferences and constraints. In reality, the disadvantage of this is of course that the employees are not working at full capacity, which increases the costs or not all employees can work at 100%.

[1] https://github.com/sgirardin/NSGA-II/tree/4.0.0

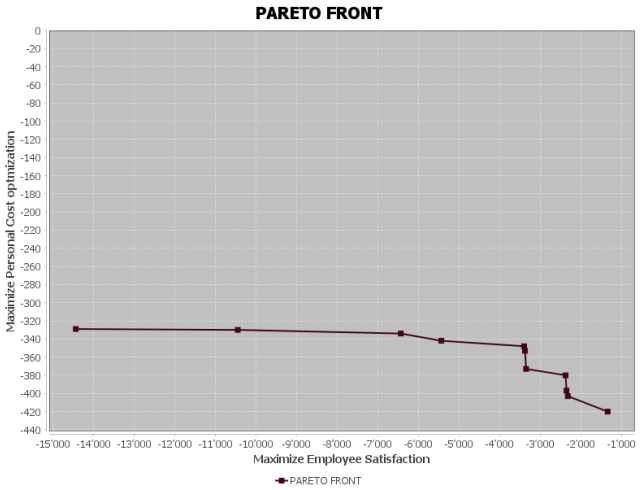

Fig. 7. Pareto front calculated with 20 employees

Another attempt to improve the algorithm we have assumed in the area of the crossover scheme. Here we changed the uniform crossover with a one-point crossover. When combining a new chromosome, the genes are not randomly selected from the parent chromosomes but crossed at one point. However, this has not improved the results.

With a Lenovo T470p laptop running on 64-bit Windows 10, which has an Intel Core i7-7700HQ CPU with 2.8GHz and 16GB RAM, the calculations could be executed as described earlier in some seconds to a few minutes. Obviously, the duration has increased if the operator number of iterations or the number of the population has been raised. A change in this mutation and crossover probability had less impact.

VI. Conclusion

In this work, the nondominated sorting genetic algorithm II (NSGA-II) was applied to optimize the train staff planning problem (TSPP) with a multi-objective approach. The chromosome representation was created using a binary coding scheme. We tested the algorithm for a theoretical small case with the objectives to minimize costs and to maximize employee satisfaction by penalizing violations of constraints and preferences. A uniform crossover and single-point mutation were executed as default. Several Parameters, namely the population size, number generations, mutation, and crossover probability have been tested to find an optimal configuration of the algorithm. It turns out, that all parameters influence the results that can be achieved. Because many hard-constraints have been violated we have tested the algorithm with more possible employees which has led to better results on the objective of the maximization of the employee satisfaction.

VII. Outlook, further work

The current solution is in some parts not very attractive because of the sometimes high penalties resulting from invalid Employee-ID's in the chromosome. We were able to prevent the invalid genes in the initial population and for mutations, but not in the crossover, thus invalid ID's can come up. One approach to prevent this would be to improve the crossover functions so that only valid ID's are produced. Another solution would be to switch from a binary coding scheme to a non-binary one, such as using a set of real numbers. To overcome these large violations caused by randomly generated gene orders in the chromosomes, a repair strategy could be used to fix all solutions decoded in the chromosomes.

Furthermore, it would be very interesting in further research to apply our proposed method to a real case. This would probably involve some additional constraints which would have to be taken into account in this specific case. The application to a larger case with a different baseline such as a long-distance train network, larger sets of services, and employees or a longer planning horizon than one week would also be interesting. It could also improve the results in regard to the violation of preferences.

VIII. References

[1] R. Domberger, L. Frey, and T. Hanne, “Single and multiobjective optimization of the train staff planning problem using genetic algorithms,” in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 2008, pp. 970–977.

[2] A. Caprara, L. Kroon, M. Monaci, M. Peeters, and P. Toth, “Chapter 3 Passenger Railway Optimization,” 2007, pp. 129–187.

[3] J. Heil, K. Hoffmann, and U. Buscher, “Railway crew scheduling: Models, methods and applications,” *Eur. J. Oper. Res.*, vol. 283, no. 2, pp. 405–425, Jun. 2020.

[4] N. Bojovic and M. Milenkovic, “Train driver rostering optimization,” in *IEEE 8th International Symposium on Intelligent Systems and Informatics*, 2010, pp. 501–504.

[5] M. Lezaun, G. Pérez, and E. S. De La Maza, “Crew rostering problem in a public transport company,” *J. Oper. Res. Soc.*, vol. 57, no. 10, pp. 1173–1179, 2006.

[6] T. Hanne, R. Dornberger, and L. Frey, “Multiobjective and preference-based decision support for rail crew rostering,” in *2009 IEEE Congress on Evolutionary Computation*, 2009, pp. 990–996.

[7] A. Ernst, H. Jiang, M. Krishnamoorthy, H. Nott, and D. Sier, “Rail Crew Scheduling and Rostering Optimization Algorithms,” 2001, pp. 53–71.

[8] L. Xie, “Metaheuristics approach for solving multi-objective crew rostering problem in public transit,” Paderborn, 2013.

[9] C. H. Chen, T. K. Liu, J. H. Chou, and C. C. Wang, “Multiobjective airline scheduling: An integrated approach,” *Proc. SICE Annu. Conf.*, no. 1, pp. 1266–1270, 2012.

[10] H. Habibnejad-Ledari, M. Rabbani, and N. Ghorbani-Kutenaie, “Solving a multi-objective model toward home care staff planning considering cross-training and staff preferences by NSGA-II and NRGA,” *Sci. Iran.*, vol. 26, no. 5 E, pp. 2919–2935, 2019.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[12] K. Hoffmann, “A Hybrid Solution Approach for Railway Crew Scheduling Problems with Attendance Rates,” 2017, pp. 243–250.

[13] A. Debabrata, “An implementation of NSGA-II in java.” GitHub repository, 2019.

IX. APPENDIX

TABLE I. CALCULATED TESTING RESULTS

Run	Parameters					Results		
	Population size	*Number of Generations*	*Mutation probability*	*Crossover probability*	*Train crew size*	*Best cost*	*Best preferences*	*number of Pareto points*
1	15	10	0.4f	0.7f	10	-422	-21661	6
2	25	10	0.4f	0.7f	10	-388	-19800	8
3	35	10	0.4f	0.7f	10	-406	-19676	9
4	45	10	0.4f	0.7f	10	-411	-18695	9
5	55	10	0.4f	0.7f	10	-388	-16812	9
6	65	10	0.4f	0.7f	10	-397	-16724	7
7	75	10	0.4f	0.7f	10	-397	-18711	7
8	85	10	0.4f	0.7f	10	-414	-18738	4
9	95	10	0.4f	0.7f	10	-398	-18727	9
10	105	10	0.4f	0.7f	10	-400	-18713	5
11	115	10	0.4f	0.7f	10	-395	-17785	11
12	125	10	0.4f	0.7f	10	-410	-17684	7
13	150	10	0.4f	0.7f	10	-378	-16784	8
14	200	10	0.4f	0.7f	10	-386	-15686	8
15	250	10	0.4f	0.7f	10	-400	-17650	7
16	300	10	0.4f	0.7f	10	-399	-14728	18
17	500	10	0.4f	0.7f	10	-390	-15731	9
18	300	50	0.4f	0.7f	10	-368	-15756	10
19	300	100	0.4f	0.7f	10	-370	-14725	13
20	300	800	0.4f	0.7f	10	-355	-13703	12
21	300	400	0.4f	0.7f	10	-376	-13662	12
22	350	400	0.4f	0.7f	10	-366	-13674	17
23	300	200	0.4f	0.7f	10	-370	-12794	12
24	300	250	0.4f	0.7f	10	-374	-13760	12
25	300	300	0.4f	0.7f	10	-364	-12737	10
26	300	350	0.4f	0.7f	10	-371	-14703	15
27	300	200	0.6f	0.7f	10	-376	-13731	16
28	300	200	0.1f	0.7f	10	-321	-11701	16
29	300	200	0.2f	0.7f	10	-353	-13604	11
30	300	200	0.3f	0.7f	10	-338	-14704	9
31	300	200	0.5f	0.7f	10	-373	-13637	9
32	300	200	0.1f	0.5f	10	-319	-11636	20
33	300	200	0.1f	0.1f	10	-314	-9729	20
34	300	200	0.1f	0.2f	10	-315	-10666	17
35	300	200	0.1f	0.3f	10	-318	-8705	16
36	300	200	0.1f	0.4f	10	-320	-9662	13
37	300	200	0.1f	0.4f	15	-332	-4518	12
38	300	200	0.1f	0.4f	20	-329	-2356	16
39	300	200	0.1f	0.3f	20	-329	-1354	11