Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 5017–5022

Order Picking Problem in a Warehouse Hospital Pharmacy

B. Beroule, O. Grunder, O. Barakat * O. Aujoulat **

* Nanomedicine Lab, Univ. Bourgogne Franche-Comté, UTBM, F-90010 Belfort, France (e-mail: benoit.beroule,olivier.grunder@utbm.fr)

** GHRMSA, Mulhouse hospital center 68000 Mulhouse, France (e-mail: aujoulato@ghrmsa.fr)

Abstract: This paper presents an optimization method for simultaneously routing improvement and items location as part of order picking operation in a warehouse of a hospital pharmacy. In this precise context, the classical order forms are replaced by medical services endowment lists which are not modified over time (or not significantly). This particularity allows new considerations and modifications of the order picking optimization methods. This study focuses on the items location determination in the warehouse and the pickers rooting in a picker-to-part low level policy context. In order to solve the considered problem, the paper details an exact mathematical solving method through a mix integer programing model which can easily be extended to a linear version. For real case, this class of model may need prohibitive computation times to be solved. Consequently, a genetic algorithm and a dedicated heuristic are presented in order to implement a hybrid genetic algorithm which is the combination of both methods. Results are provided to highlight the contribution of the hybrid method compared to the others in terms of computation times and solutions quality.

Keywords: hospital logistics ; warehouse ; order picking ; genetic algorithm

1. INTRODUCTION

Warehouse management optimization is a widely studied research field and especially the order picking operation. Indeed, in 2003 the part due to warehousing in companies logistics costs was evaluate to 20 % in average (Kearney and Kearney (2004)). Furthermore, order picking is commonly the most expansive operation of warehouse management with an average of 50 % to 75 % (Coyle et al. (1996)). Scientific literature shows that order picking operation is composed of several aspects which can be improved such as: picking, layout design, storage assignment, zoning, batching or routing (De Koster et al. (2007)). Unlike the classical industrial warehouses, in a hospital, the order forms are dependent of medical services and may be known in advance. This particularity allows considering new warehouse management methods which, as our knowledge, have not been really studied for now. In a classical industrial warehouse, the order picking operation optimization methods are most of the time based on the traveling salesman problem (TSP). (Ratliff and Rosenthal (1983)) used a TSP algorithm to minimize order picking operation duration in a warehouse organized in linear aisles with crossovers, the result may be computed in a reasonable time for fewer than 50 aisles. In order the minimize the computation time, Achuthan et al. (1996) determined a new sub-tour elimination constraint for the TSP (and even extended to the vehicle routing problem (VRP)). This new constraint implied to compute numerous intermediate results before finding the optimal solution but drastically decreased the global computation time. This new model allows solving problems with more and more storage locations. Consequently, numerous papers deal with approximate or exact methods inspired by the TSP in order to solve the picking order problem. (Mohsen and Hassan (2002)) developed a framework for the design of warehouse layout which determines the amount and location of docks and input/output points as well as the number of aisles, their dimensions, orientations and other parameters. The framework takes into account lots of warehousing aspects to provide the best possible layout depending on performance targets. (Moeller (2011)) focused on the picking sequences obtained by the TSP algorithms by divided them into three categories. Actually, a sequence may be determined by considering the global warehouse system, the picker or the mathematical TSP model, the study highlights indeed the fact that sometimes a pickers may considered a picking sequence as weird and modify it. In some cases, the exact methods are not sufficient to solve the order picking problem (numerous aisles, specific constraints, real time computation), this is why approximate methods may be used to save computation time. (CHANG et al. (2007)) implemented an order picking operation mathematical model as well as a genetic algorithm to solve it. It appears in the literature that population based algorithm provided interesting results when determining order picking. (Hsu et al. (2005)) proposed a genetic algorithm to provide an order picking batching method, (Jianqin and Kongjie (2004)) implemented a hybrid method composed of a genetic algorithm and a neural network performing high speed local search. Other population based algorithms are used, (Önüt et al. (2008)) presented a particle swarm

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.910

optimization-based algorithm with a constriction factor to solve a multi-level warehouse layout design problem. This study is particularly focused on routing and storage assignment aspects in the particular context of hospital warehouses. These aspects must be considered together and not one after the other to ensure determining the best possible solution. In this paper, the case of a hospital warehouse is detailed through several parts. In a second part the precise context of the study is presented. Then, in a third part the mathematical model is implemented and discussed. In a fourth part several solving methods (heuristics or meta-heuristics) are described. Finally, the fifth part presents some results to compare the methods in terms of computation time and solution quality to finally conclude on their efficiency.

2. CONTEXT OF THE STUDY

Generally speaking, a hospital center is composed of several sectors. This work is focused on the pharmacy which is the place where the warehousing management aspects are the most crucial.

2.1 The Pharmacy

This paper studies the order picking operation in a hospital pharmacy context. Indeed, within a hospital center, the pharmacy is the place (most of the time a separate building) where the medicinal products and the surgical devices are managed and stored in a dedicated warehouse. This particular context implies precise constraints and characteristics which may differ from industrial warehouses. In this study, a picker-to-part low level policy is considered, meaning that the pickers must move on foot or in a vehicle until they reach the right place to perform the pick at the floor level. Then, after performing the pick, they move to the next item (according to an order form) or go back to the depot. It should be borne in mind that the work may be easily extended to other picking policies. To determine the lists of items which have to be picked, each hospital service provides an order form to the pharmacy, in this context they are called endowment lists. It contains the endowments needed by the service for a given duration (a week, a month). Unlike an industrial warehouse, the pharmacy order forms are almost always identical (or slightly different) from one year to the next. This is due to the fact that the needs of a hospital service do not change significantly over time. This particular aspect allows reconsidering the pharmacy order picking optimization problem. In this study, the objective is to minimize the average time needed by the pickers to complete the hospital services endowment lists. The particular aspect of this work lies in the optimization factors. Actually, here we are looking for the best items localisation and the best listing order for each endowment list. These two aspects must be considered and optimized together to ensure determining the best solution.

2.2 Problem Scope

The problem in this case is to implement hybrid optimization methods that take into account simultaneously the routing of the pickers and the location of the items in the pharmacy in order to optimize the picking operation. These two aspects must be considered simultaneously to ensure the finding solution to be optimal. Considering items to store in the pharmacy, possible locations for these items and endowment lists (composed of items), the problem consists in determining a item-location matching (meaning placing each item in one and only one possible location) and ordered endowment lists to minimize the distance traveled by the pickers when performing the picking operation of each list in average. By minimizing the traveled distance the duration time is minimized as well.

The obtained solution only depends on: the number of items, the number of endowments lists and their contents and the distance between the proposed items locations.

3. MODELLING

The problem detailed above needs to be mathematically modeled in order to determine the optimal solution. To avoid unnecessary complexity, the following assumptions are made:

- The collisions between pickers are not considered.
- The pickers (or picking vehicles) are never overloaded.
- Picking operation starts and ends at the same place (the depot).
- A class of items must be placed at only one location.

In the case of a warehouse with large aisles or with only one picker, the "no collisions" simplification does not impact the results. Otherwise, the different paths of each picker should be considered and a time penalty might be added when two of them met. Moreover, considered a maximum storage capacity of the picker to be overloaded implies to perform several rounds in the warehouse in order to complete a picking list. This aspect is not considered in this study for clarity reasons. Furthermore, the parameters must allow to represent the different aspects of the routing and the items location. Consequently, the model parameters and decision variables are defined as follows.

Parameters

- n : number of items (and locations).
- m : number of picking lists.
- C_t : set of items contained in list t.
- $D_{k,l}$: distance between location k and l.

Decision variables

- $x_{i,k}$: binary variable equal to 1 only if item i is at location k.
- $l_{i,j,t}$: binary variable equal to 1 only if list t implies to go from item i to item j when performing picking operation.
- $u_{i,t}$:the picking order of item i in list t

The aim here is to minimize the distance traveled by the pickers when performing the order picking operation. This is why we defined the objective function as follows.

$$Minimze\ z = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\sum_{t=1}^{m} x_{i,k}.x_{j,l}.l_{i,j,t}.D_{k,l} \quad (1)$$

In this equation, the variable z represents the sum of the distances between each items that have to be picked consecutively. Actually, the expression "$x_{i,k}.x_{j,l}.l_{i,j,t}$" is equal to 1 only if $x_{i,k} = 1$ meaning that item i is located on location k, $x_{i,k} = 1$ meaning that item j is located on location l then finally $l_{i,j,t} = 1$ meaning that there is a list t that contains item i and j consecutively. If all of these constraints are satisfied, then the picker will indeed traveled from location k to location l sometime during the picking operation and then the distance between these two locations ($D_{k,l}$) has to be added to z. It is important to notice that here the objective function is not linear but, classical binary values product linearization methods may be used to obtain a MILP model, the linear version is not included in this paper for reasons of clarity. Moreover, some constraints are considered.

$$\sum_{j=1}^{n} x_{i,j} = 1 \qquad 1 \leq i \leq n \tag{2}$$

$$\sum_{i=1}^{n} x_{i,j} = 1 \qquad 1 \leq j \leq n \tag{3}$$

$$x_{1,1} = 1 \tag{4}$$

$$\sum_{j \in C_t} l_{i,j,t} = 1 \qquad i \in C_t \quad 1 \leq t \leq m \tag{5}$$

$$\sum_{i \in C_t} l_{i,j,t} = 1 \qquad j \in C_t \quad 1 \leq t \leq m \tag{6}$$

$$l_{i,i,t} = 1 \qquad 1 \leq t \leq m \tag{7}$$

$$u_{1,t} = 1 \qquad 1 \leq t \leq m \tag{8}$$

$$u_{i,t} - u_{j,t} + n l_{i,j,t} \leq n - 1 \qquad \begin{array}{r} 1 \leq i \leq n \\ 2 \leq j \leq n \\ 1 \leq t \leq m \\ i \neq j \end{array} \tag{9}$$

Equations (2) and (3) force each item to be placed in a unique location and a location to be occupied by a unique item respectively. In this study, the location number 1 is considered as the depot, meaning the place where the picker will start and end the picking operation. This is why equation (4) prevents the depot from being placed elsewhere than at location number 1. The equations (5) and (6) represent the fact that in any list, each item is preceded and followed by a unique item respectively. It has to be noted that the lists are considered as cyclic. Indeed, each of them starts and ends with the item number 1 (the depot). The equation (7) is used to prevent a list from containing the same item twice and consecutively. Finally, the equations (8) and (9) prevent the lists from containing sub tours. These equations allow to precisely model the studied problem. They can be solved with an exact solver but the complexity of the objective function implies a drastic increase in computation time. Indeed, the computation time may reach an hour with only 10 wares. This is why other approximate methods are needed to determine acceptable solutions in reasonable computation time.

4. SOLVING METHODS

Several methods may be considered to solve the studied problem. As seen before, it appears in lots of studies that the genetic algorithms provide good results when solving these class of problems (Grefenstette et al. (1985); Larrañaga et al. (1999)). Moreover, in order to determine a solution as optimized as possible, a hybrid genetic algorithm is implemented. The hybrid genetic algorithm combines several optimization methods in order to provide a good solution. It is composed of two steps. First the dedicated heuristic is used on the set of considered picking lists to obtain an items location minimizing the picking duration according to these lists. Then, a TSP algorithm may be used on each picking list based on the solution determined before to order each of them. A solution is then created, it is composed of the previously determined items location and the picking lists ordered according to the TSP algorithm solution. Finally, a genetic algorithm is initiated with a population composed of randomly created solutions and the previously founded solution before being computed. The output is considered as the solution provided by the hybrid genetic algorithm. The figure (1) summarizes this process.

4.1 Genetic Algorithm

The genetic algorithm is a meta-heuristic invented in 1960 by John Holland and popularized by David Goldberg (Golberg (1989)). It is based on the evolution mechanisms of a populations. A population composed of potential individual solutions is created and, cycle after cycle, new solutions are created by crossing and mutating the individuals. To implement a genetic algorithm the first step consists in determining a solution representation. Here each solution is represented as in the equations (10).

$$P = \begin{cases} L = \{L_1, L_2, ..., L_m\} \\ X = [X_1, X_2, ..., X_n] \end{cases} \tag{10}$$

Each solution S is composed of a set of ordered picking lists L and a location vector X. Each list L_i is an items vector which respects a precise order the picker will used to perform the picking operation. The location vector X determines the matching between each item and its location, the item number i is placed at location number $X(i)$. The genetic algorithm details in this paper starts by creating an initial population composed of randomly created solutions. Then, these primary solutions evolve during several generations (or cycles) through two genetic mechanisms, the cross-over and the mutation. The cross-over is a genetic mechanism which allows parent solutions to be merged to create a child solution. During each cycle, a part of the current population is selected (according to a determined crossing rate) to be crossed by pairs. It is performed thanks to two crossing operators.

The first operator identifies the robust parts of each list L_i and L'_i of two parent solutions (meaning the items that are placed at the same position in the list L_i and L'_i) and then create a new child solution composed of lists containing the same robust items. The non-robust items are randomly selected and the location vector X of one of the two parents is randomly kept for the child solution. The second cross-over operator is similar to the previous one but identifies the robust parts of the location vector X and X' (and the list set L of one of the two parents is randomly kept for the child solution). When two solutions have to be crossed, a cross-over operator is randomly chosen. An example of these two operators is shown on figures (11) and (12). When two solutions have to be crossed, then a crossing operator is randomly chosen.

$$\left.\begin{array}{l} L_i = [2, 5, 8, 6, 7, 4] \\ L'_i = [8, 5, 6, 2, 7, 4] \end{array}\right\} \rightarrow [6, 5, 2, 8, 7, 4] \tag{11}$$

$$\left.\begin{array}{l} X \ = [1, 4, 6, 2, 3, 7, 5, 8] \\ X' = [1, 6, 7, 2, 8, 4, 5, 3] \end{array}\right\} \rightarrow [1, 7, 3, 2, 4, 6, 5, 8] \tag{12}$$

The mutation is a genetic mechanism which allows a solution to be slightly modified by itself. During each cycle, a part of the current population is selected (according to a determined mutation rate) to be mutated. It is performed thanks to two mutation operators. The first operator randomly select two items in a randomly selected list of the lists set L of the solution and then exchange them. The second operator randomly select two elements of the location vector X of the solution and then exchange them. When a solution has to be mutated, a mutation operator is randomly chosen. An example of this two operators is shown on figures (13) and (14).

$$L_i = [2, 5, 8, 6, 7, 4] \longrightarrow [2, 6, 8, 5, 7, 4] \tag{13}$$

$$X = [1, 4, 6, 2, 3, 7, 5, 8] \longrightarrow [1, 3, 6, 2, 4, 7, 5, 8] \tag{14}$$

When a part of the population has been crossed and mutated, each solution is evaluated (from former and new generation) and the best ones are selected to create a new population of the same size than the initial. A new cycle is then started until the predetermined number of cycle is reached. The best solution of the final generation population is considered as the output of the genetic algorithm.

4.2 Initial solution

In order to improve the efficiency of the genetic algorithm and create the hybrid genetic algorithm, an initial solution is computed with a dedicated heuristic. The purpose of the dedicated heuristic is to intuitively construct an interesting solution. This solution may be therefore improved (for instance by a local search). The dedicated heuristic implemented in this paper is composed of three steps.
Step 1 : Each picking list is considered and the number of occurrences of each item is counted. The items are then sorted from the most recurrent to the less recurrent and the result is represented by a vector V ($V(1)$ is the item which appears most of the time in the lists then $V(2)$ is the second one and so on).
Step 2 : The possible locations of the items in the warehouse are sorted according to their distances to the depot (from the nearest to the farthest). Then, the result is represented by a vector V' ($V'(1)$ is the nearest location from the depot and so on).
Step 3 : Each item is then allocated to a location depending on their place in the vector V and V', the item $V(i)$ is located at the place $V'(i)$.

This method allows to stored most picked items as near as possible from the depot. It seems intuitively to provide good solutions (the quality of the solutions will be discussed afterwards). It is worth noting that the heuristic does not modify the picking list order but only the wares location.

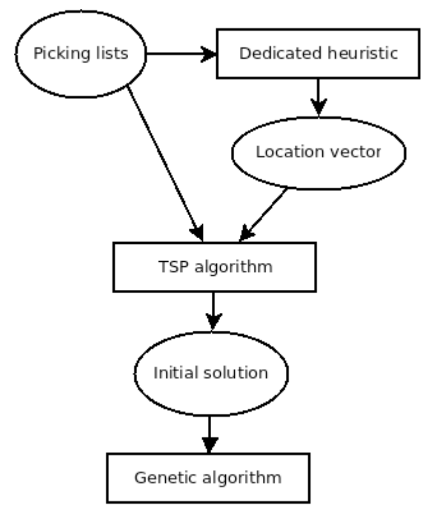

Fig. 1. Hybrid genetic algorithm

5. RESULTS

In order to evaluate the relevance of the proposed solving methods, they have to be compared in terms of solutions fitness and computation times. therefore, each previously detailed method (genetic algorithm, dedicated heuristic and hybrid genetic algorithm) is used to solve a set of scenarios. The experiment is conducted as follows.
Every scenario is composed of a warehouse and a set of picking lists. The warehouse is identical for each scenario and is composed of 400 items locations. The number of picking list is between 1 and 25 and in each case the number of items in the lists is set to a value from 10 to 90 with a step of 10. The set of scenarios thus created is solved with each solving method. The genetic algorithm is parametrized as follows.

- $initialpopulation = 200$
- $numberofcycles = 1000$
- $crossingrate = 50\%$
- $mutationrate = 30\%$

The table of the appendix (A) detailed the results obtained in each case. The results are composed of the fitness of the best solution (meaning the distance traveled by the picker in distance unit) and the time needed to compute it (in seconds). In every case, the computation time limit is fixed to one hour. When the time limit is reached before the end of the considered algorithm, the best currently founded solution fitness is then provided. In the case of the hybrid genetic algorithm, the time limit is fixed to one hour for the dedicated heuristic and one hour for the genetic algorithm. Therefore it is important to note that each result (fitness and computation time) is an average on 10 different scenarios sharing the same properties (same amount of lists and items in the lists).

This results table leads to several remarks.
The improvement ratio of the hybrid genetic algorithm compare to simple genetic algorithm ($\Delta_{HGA/GA}$) and compare to the dedicated heuristic ($\Delta_{HGA/DH}$) are computed according to equations (15) and (16) respectively.

$$\Delta_{HGA/GA} = \frac{Fit_{HGA} - Fit_{GA}}{Fit_{GA}} \times 100 \tag{15}$$

$$\Delta_{HGA/DH} = \frac{Fit_{HGA} - Fit_{DH}}{Fit_{DH}} \times 100 \tag{16}$$

In average on each scenario $\Delta_{HGA/GA} = 23\%$ with a maximum of 38.1% and a minimum of 9.4% and in average $\Delta_{HGA/DH} = 30\%$ with a maximum of 52% and a minimum of 0% (meaning that the GA do not improve the initial solution). Consequently, we observe that the hybrid genetic algorithm is very efficient compare to classical genetic algorithm even if no comparison with exact solutions can be made. Indeed, the high complexity of the objective function (see equation (1)) prevent the problem from being solved with exact solvers in real case because of prohibitive computation times.

It should be noted that no comparison with the MILP model results are maid. Actually, as said before, the MILP model can only be used to solve small instances because of the problem complexity. Unfortunately, with these kind of instances the genetic algorithm always provides the best solution and then prevent the comparison.

6. CONCLUSION

The pharmacy warehouse management case implies new considerations compared to a classical industrial case. The purpose of this paper is to present a method to improve the pharmacy logistic by providing decision tools to decide both the items location in the warehouse and the picking order of the medical service endowment list. Unfortunately, this problem appears to be too complex to compute exact solutions and this is why approximate methods must be considered. The hybrid genetic algorithm is a combination of a meta-heuristic (genetic algorithm) and a dedicated heuristic and provides efficient solutions in terms of picking time duration. These results still need to be completed with other heuristics, meta-heuristics and exact solutions. It might be in deed possible to solve larger size problems with exact methods in precise configuration of layout which have to be determined. With this work, we hope to provide efficient tools to improve the global logistic of the hospital sector which are becoming ever more important as the human life expectancy grows steadily and consequently the number of hospital stay per person.

REFERENCES

Achuthan, N., Caccetta, L., and Hill, S. (1996). A new subtour elimination constraint for the vehicle routing problem. *European Journal of Operational Research*, 91(3), 573–586.

CHANG, F.l., LIU, Z.x., Zheng, X., and LIU, D.d. (2007). Research on order picking optimization problem of automated warehouse. *Systems Engineering-Theory & Practice*, 27(2), 139–143.

Coyle, J.J., Bardi, E.J., Langley, C.J., et al. (1996). *The management of business logistics*, volume 6. West Publishing Company St Paul, MN.

De Koster, R., Le-Duc, T., and Roodbergen, K.J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2), 481–501.

Golberg, D.E. (1989). Genetic algorithms in search, optimization, and machine learning. *Addion wesley*, 1989, 102.

Grefenstette, J., Gopal, R., Rosmaita, B., and Van Gucht, D. (1985). Genetic algorithms for the traveling salesman problem. In *Proceedings of the first International Conference on Genetic Algorithms and their Applications*, 160–168. Lawrence Erlbaum, New Jersey (160-168).

Hsu, C.M., Chen, K.Y., and Chen, M.C. (2005). Batching orders in warehouses by minimizing travel distance with genetic algorithms. *Computers in Industry*, 56(2), 169–178.

Jianqin, T.G.Z.P.Y. and Kongjie, L.F.S. (2004). Research on the fixed shelf order-picking optimization problem using a kind of hybrid genetic algorithm [j]. *Chinese Journal of Mechanical Engineering*, 2, 030.

Kearney, E. and Kearney, A. (2004). Excellence in logistics 2004. *European Logistics Association, Brussels.*

Larrañaga, P., Kuijpers, C.M.H., Murga, R.H., Inza, I., and Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13(2), 129–170.

Moeller, K. (2011). Increasing warehouse order picking performance by sequence optimization. *Procedia-Social and Behavioral Sciences*, 20, 177–185.

Mohsen and Hassan, M. (2002). A framework for the design of warehouse layout. *Facilities*, 20(13/14), 432–440.

Önüt, S., Tuzkaya, U.R., and Doğaç, B. (2008). A particle swarm optimization algorithm for the multiple-level warehouse layout design problem. *Computers & Industrial Engineering*, 54(4), 783–799.

Ratliff, H.D. and Rosenthal, A.S. (1983). Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. *Operations Research*, 31(3), 507–521.

Appendix A. SOLVING METHODS COMPARISON

Table A.1. solving methods comparison

		Dedicated Heur. + TSP		Hybrid GA		GA		Improvement (%)	
lists	items	fit.	time(s)	fit.	time(s)	fit.	time(s)	$\Delta_{HGA/GA}$	$\Delta_{HGA/DH}$
1	10	1060	8	1060	36	1545	29	31.4	0.
1	20	1926	59	1804	100	2860	41	36.9	6.3
1	30	3385	3600	3073	3649	4444	49	30.9	9.2
1	40	4904	3600	4177	3657	5614	58	25.6	14.8
1	50	6485	3600	5357	3665	7050	64	24.	17.4
1	60	7203	3600	6245	3672	8822	71	29.2	13.3
1	70	8940	3600	7015	3679	10240	80	31.5	21.5
1	80	14368	3600	8758	3686	12631	89	30.7	39.
1	90	21224	3600	10158	3694	13982	96	27.3	52.1
5	10	8696	101	7391	165	8844	67	16.4	15.
5	20	22127	3602	15531	3709	18348	115	15.4	29.8
5	30	37416	3602	24421	3748	29539	154	17.3	34.7
5	40	54960	3602	32690	3834	40541	195	19.4	40.5
5	50	56662	3602	40385	3831	53922	238	25.1	28.7
5	60	84730	3602	50890	3866	67877	276	25.	39.9
5	70	101749	3602	60895	3904	86687	315	29.8	40.2
5	80	102707	3603	68190	3949	100373	363	32.1	33.6
5	90	128855	3603	79135	3989	118794	406	33.4	38.6
10	10	22542	234	17645	342	20722	113	14.8	21.7
10	20	62193	3605	39088	3798	44302	199	11.8	37.2
10	30	115841	3605	61663	3883	71714	288	14.	46.8
10	40	141824	3605	85189	3966	102606	377	17.	39.9
10	50	171555	3605	106334	4053	137216	462	22.5	38.
10	60	225995	3605	133208	4134	175204	548	24.	41.1
10	70	233887	3605	150840	4207	218162	626	30.9	35.5
10	80	239012	3606	166657	4302	261872	708	36.4	30.3
10	90	259771	3607	186186	4373	301000	783	38.1	28.3
15	10	38581	342	30615	488	34068	151	10.1	20.6
15	20	110108	3606	68244	3890	75692	287	9.8	38.
15	30	167352	3607	104941	4015	121700	419	13.8	37.3
15	40	202642	3606	143127	4142	178876	555	20.	29.4
15	50	258144	3607	179265	4280	238574	667	24.9	30.6
15	60	302999	3608	213356	4391	301646	784	29.3	29.6
15	70	385944	3608	256273	4507	379022	916	32.4	33.6
20	10	57992	502	44395	700	50416	208	11.9	23.4
20	20	158138	3609	101259	3980	110444	376	8.3	36.
20	30	229795	3610	156281	4149	179014	550	12.7	32.
20	40	328497	3610	213846	4329	259867	705	17.7	34.9
20	50	337960	3611	252263	4505	350371	883	28.	25.4
20	60	381092	3609	292073	4663	446043	1055	34.5	23.4
20	70	563936	3611	382849	4835	549178	1222	30.3	32.1
20	80	586394	3610	419194	5047	668304	1417	37.3	28.5
20	90	636143	3612	468137	5224	778514	1602	39.9	26.4
25	10	79319	573	61445	814	69239	244	11.3	22.5
25	20	224483	3610	136871	4089	151096	481	9.4	39.
25	30	312402	3613	211028	4310	241867	693	12.8	32.4
25	40	398775	3613	277197	4536	349975	913	20.8	30.5
25	50	436261	3614	330577	4759	472986	1110	30.1	24.2
25	60	637110	3614	336129	3722	455753	999	26.2	47.2