IoT Based Smart Staff Allocator in Quick Service Restaurants

Kerem AYTAÇ
Marmara University,
Ata Technology Platforms (ATP) R&D Center
Istanbul, Turkey
kerem.aytac@marun.edu.tr

Ömer KORÇAK
Marmara University
Istanbul, Turkey
omer.korcak@marmara.edu.tr

***Abstract*—Internet of Things (IoT) term has been a trend topic for a long time, and takes over many jobs from human-controlled domains and makes the things easier, quicker and remotely controllable with smart automations. Quick service restaurants include many parts that human intervention is involved, and burns so much effort that must be well organized and automated. It is such an era that an effort must be passed to IoT-brains where possible, and human should pay the gained effort to any other areas. Quick service restaurants have many staffs, especially at the back office, created by kitchen, storage etc. These staffs must be well and efficiently organized so that there must not be a waste of effort. A human brain might not be sufficient for this duty and also it will be costly. So, it will be controlled by an IoT brain which is fed by many sensors within restaurant and distribute the jobs to them fairly, efficiently and less-costly. In this paper, we present an architecture for allocating jobs to staffs and tracking their performance for various tasks. We also propose and evaluate a genetic algorithm with novel selection method in order to solve the task assignment problem.**

I. INTRODUCTION

Quick Service Restaurants (a.k.a Fast-Food) are attracting much more people since they provide low-cost and time-efficiency. Every year the number of people who prefers Quick Service Restaurants increasing dramatically. The global revenue of quick service restaurants converges to a trillion of dollar year by year, which is bigger than the total economic value of most countries. Fast food industry employs millions of people, which means a huge amount of human source is dedicated to this sector [1]. Managing this vast amount of swarm requires hundreds of thousands of people which corresponds to a great amount of outgoing for restaurants.

Not only managing them is hard, but also finding the weakest links to get rid of or fixing them or the strongest ones to award with incentives etc. Some key staffs exist in restaurants to orchestrate this staff community, and dedicates themselves for them. The more they pay time with the same community, the easier it becomes to orchestrate. However, when that orchestrator leaves for any reason, then it becomes a nightmare for the restaurant owner. Because adopting a new key staff and making compatible requires many time and effort with some waste of efficiency. This process should be exempted for human control and must be taken over by IoT brain. An IoT brain is not like human, it easily gets used to a new link in a chain with some basic inputs in a little time. It never shows mercy and it gives only what it is deserved. Because it has no senses, but only numeric inputs and solid decisions.

Many staff allocating works have been shown up so far. One example is from another domain, a shift planner of nurses by scheduling them and find a cure for specific type of problems that occurs in that domain [3]. It is called as staff scheduling and it is the assignment for staffs to time shift slots by obeying many constraints. In this project, they point to a specific problem of scheduling nurses on daily shifts for a month of calendar. That solution tries to assign the related shifts with some constraints like a specific number of nurses on a pre-defined shift, a requirement of work for every third weekend by avoiding overtime works and working of three consecutive workdays as much as possible. In this study, they process two genetic algorithm based staff-scheduling solutions for scheduling nurses at a hospital. First solution drives on a traditional way, which is bit-string chromosome structure, the other one, uses a two-dimensional array chromosome structure to represent each schedule.

Another work presented in [6] schedules the staff with mixed skills. Having three main objects within the work, the first one is minimizing the total cost of assigning staff to satisfy the manpower requirement over time. The second one is, to extract a solution with the max surplus of staff in solutions with nearly same level of assigning cost. For the last objective, it tries to reduce the variation of staff surplus over different scheduled periods. It also proposes a new genetic algorithm method to solve these three objective problems. It approaches from other angle comparing with traditional genetic algorithm with three components at coding scheme (gene order in chromosome), parent selection and crossover phase.

This paper is based on an IoT based edge computing architecture proposed and developed for Quick Service Restaurants, which is presented in [2]. In the proposed architecture, there are many sensors connected to an edge gateway, which sends valuable information to the cloud as well as taking some instantaneous actions in an intelligent manner. In [2], we presented several smart modules, such as intelligent weight-meter and production service level estimation, for waste management and service optimization. **Intelligent weight-meter** module is responsible for sensing the weight of waste

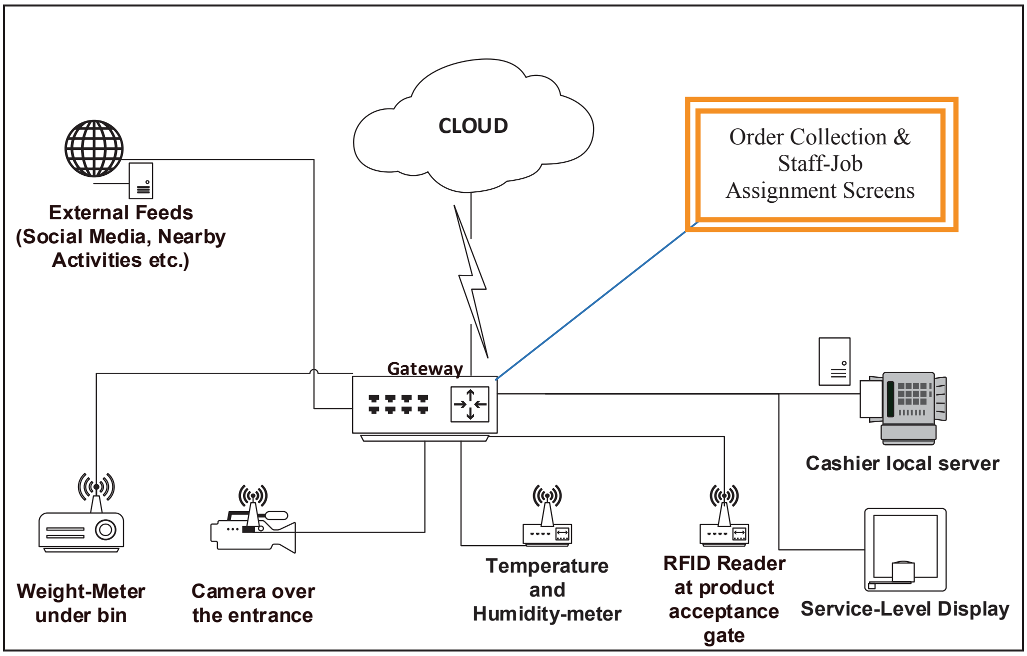

Fig. 1. Edge Oriented IoT Architecture for a Quick Service Restaurant, examine-candidate device for this paper is highlighted [2]

bin and finding out the anomalies and the trend of usage. It tries to initiate some alerts and algorithms for the anomalies or the trend of usage. Basically it aims to reduce and manage the waste. On the other hand, **production service level estimator** module which is integrated with device level display, briefly identifies the current value of production fulfillment level. In other words, it scores the current level of production workload density from one to seven. Scoring one means production workload depends on only when an order arrives, meaning that reactive production is enough. Scoring seven means that it is a very busy time and heavy proactive production workload is necessary together with the reactive production. These levels are decided by gateway by running some machine learning algorithms trained by many values by various sensors.

This paper focuses on another smart module in the context of Quick Service Restaurant automation, namely **smart staff allocator**. This module deals with a novel problem in the domain of optimal job assignment which can be considered as a subset of service optimization. It is integrated to staff-job assignment screen. This screen device has a reflection in any restaurant somehow. If you keep it simple, it simply shows the order list to be prepared. In some restaurants, orders are written on receipts or papers and pinned in front of production staff, which is the primitive way. Some restaurants have basic order screens. Staffs follow for what to prepare on that screen. In this restaurant, not only orders, but any tasks can be followed. To be realistic, a staff in a kitchen is not only supposed to prepare only food, but also carry some boxes from/into cold storage rooms, clean dishes or kitchen, deliver some item to somewhere or someone, many things you can imagine. There are many drawbacks if they use the simple order tracking screen and job distribution system. For instance, manager watches the group, and detects an idle staff and may tell him/her to carry that box regardless of how he/she is good at it or how strong he/she is. As another instance, when a complicated or profession-required order arrives, you need ‘that’ key staff to prepare it. But if you waste him to other basic, or non-productive jobs, then that order will have to wait.

Since time and efficiency is extremely important in the massive fast food sector, intelligent allocation of jobs is a crucial problem. However, there is not much work in this important area. Major contribution of this work is to fill this gap by providing a novel solution approach based on genetic algorithm. The rest of the paper is organized as follows. Next section describes system architecture and gives some details of test environment. Section III gives description of smart staff allocation problem. Section IV presents solution approach based on genetic algorithm and provides results for various parameters. Section V concludes the paper.

II. System Architecture and Problem Definition

A. System Overview

The edge-oriented IoT architecture proposed and developed for Quick Service Restaurants is illustrated in Figure 1. Various sensors with various responsibilities are connected to an edge gateway. Gateway is the bridge between all devices. Intelligence is located in the gateway. Gateway preprocess data received from sensors, it eliminates the redundant ones and apply some machine learning or optimization algorithms to take some intelligent decisions. For smart staff allocation, the gateway gets order and task information from order collecting screens, cashier local server and some authorized devices. After running optimization algorithm, it sends the results to staff-job assignment screen. The input and output devices for smart-staff allocation is shown in Fig. 2 in a closer view. Besides the order collecting screens, manager’s or any other key staff’s phone will also infiltrate into queue with task assignments. It can be any task to be fulfilled. Staff-Job assignment screen displays the assignment results. It is also equipped with readers so that it knows when a staff takes over and completes a job. More details are given in Section III.A.

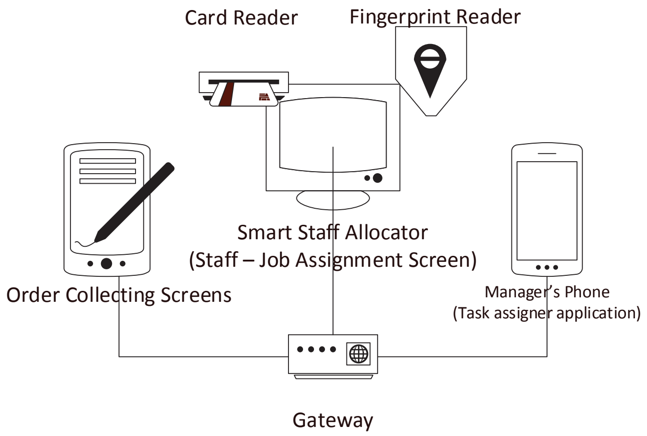

Fig. 2. Smart Staff Allocator Module in the Network

B. Test Environment

For the test domain, a famous quick service restaurant chain is picked that produce hamburgers in Turkey with more than 500 restaurants spread to country. A Raspberry Pi 3 microcontroller is dedicated to its precious duty as a gateway with an embedded Windows IoT Core. All sensors shown in Fig. 1 are connected to the gateway. Sensors generally use wireless protocols such as Wifi, Bluetooth or ZigBee, which is a good choice for communication in IoT. ZigBee protocol has low-frequency signals with a good cover range and low energy consumption [4]. Mobile phone uses local area network to communicate with gateway. If needed, it can use wide area network when out of restaurant. It is just matter of network setup. It has specifically designed mobile applications inside, and with drag-drop actions, managers can assign tasks to the current screen while monitoring the current queue. They can dequeue, requeue a task as well.

III. Smart Staff Allocation

A. Problem Definition

In most of the contemporary restaurants, there is an order queue screen for the staff, and they assign the orders to themselves or assigned by a manager. However, these assignments happen in a blink of an eye, and decided by human. So, allocation would not be optimal. In addition to this, it would be allocation waste. Sometime, these assignments create unfairness due to heterogeneous distribution, that is, some staff have five orders to prepare where another one have only two. So we need gateway to collect all the task and order information, and assign tasks to staff in a fair and efficient way.

The main objective is to minimize the total cost, while providing fairness. Cost of an assignment depends on proficiency of a staff on performing that task and also difficulty level of that task. In addition to this, an additional delivery penalty is incurred depending on the type of an order. These costs and penalties are clarified in the test case given in the next subsection.

In the IoT based system, gateway collects any orders from the cashier. Order collection screen has also an RFID reader or integrated with fingerprint device which knows who is working on that shift. With RFID concept, any workers will be given bracelets to their wrist. So when a staff takes over the job from the screen by touching, it will detect the closeness of the hand, and the RFID tagged bracelet as well. In other way, staff can read their fingerprints before touching the related task on the screen. So that, our screen knows the staff and the capabilities of the staff. Staffs can be classified and scored by managers within regular intervals such as, a specific staff is good at preparing product A, but not product B or if a staff is newbie, he/she can prepare easily product C, but not fast and good at the others. Besides this classification by managers, gateway can also find out how he/she is good at preparing that product from a time on. Because, staffs mark as complete the order when finished. So gateway compare the performance with the others, and if it is a slow preparation, staff will be low-scored for that product. These scores can be evaluated and manipulated by managers in anytime. If a staff has a low-score for product, then that products preparation by that staff incurs high-cost.

B. Test Case

Let us say that we have five types of products. Twenty of orders are queued and will be distributed to five of staffs that are ready to welcome the orders. Table I shows the capabilities of these staffs.

TABLE I. Staff Proficiencies At Products

Staff Number	Capability
1	Normal for all product types
2	Bad at Product #4 (2x slower), normal for others
3	Very bad at Product #1 and #2 (3x slower), good at the others (2x faster)
4	Good at Product #0 and #1 (2x faster), fairly bad at others (0.25x slower)
5	Excels at Product #0 (4x faster), very bad at others (3x slower)

In addition to these costs, a delivery type also affects and has a penalty on it due to their importance and multiplies the cost. Because an "In Restaurant" order is very important and has to be produced immediately. So if a low-profile staff tries to produce it, its penalty multiplies it with a higher number than the others, and cost results higher than the experienced staff. Table II gives penalty coefficients for three types of delivery: in restaurant, home delivery and proactive production.

The cost of producing a product is given as the same with the product type number. For example, Product 1 is the easiest to produce with cost one, Product 5 is the hardest to produce with cost five.

TABLE II. DELIVERY PENALTIES

Delivery Type	Penalty (Applied on total cost)
0 (In Restaurant)	3x
1 (Home Delivery)	2x
2 (Proactive Production)	1x

Table III shows the product type and delivery type for each of the twenty orders.

TABLE III. ORDERS AND DETAILS

Order No	Product Type	Delivery Type
0	3	2
1	1	1
2	3	0
3	1	0
4	4	0
5	0	1
6	0	2
7	4	2
8	3	2
9	3	2
10	1	2
11	2	1
12	3	2
13	3	0
14	1	1
15	3	0
16	3	2
17	0	2
18	4	0
19	2	1

In order to provide fairness, all staff should be assigned with same amount of products, if possible. Therefore the problem is how to fill the question marks given in Table IV. In other words we have to decide on which order number should be written where. In the next section, we describe solution approach for this problem. We also present some basic results for this test case.

The overview of the full workflow is presented in Fig. 3. All lifecycle can be seen at a glance.

TABLE IV. QUESTION MARKS WILL BE FILLED AFTER THE SMART STAFF ALLOCATION ALGORITHM EXECUTES

Stf 1	Stf 2	Stf 3	Stf 4	Stf 5	Stf 1	Stf 2	Stf 3	Stf 4	Stf 5
?	?	?	?	?	?	?	?	?	?
Stf 1	Stf 2	Stf 3	Stf 4	Stf 5	Stf 1	Stf 2	Stf 3	Stf 4	Stf 5
?	?	?	?	?	?	?	?	?	?

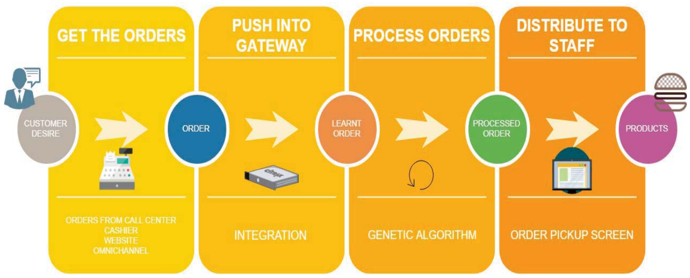

Fig. 3. Overview of the entire order process in quick service restaurant

IV. SOLUTION APPROACH

A. Methodology

The problem is a sort of multi-task generalized assignment problem which is proven to be NP-hard [11]. Therefore we consider using Genetic Algorithm which is a heuristic search inspired by the evolutionary ideas of natural selection and genetics. Genetic algorithm uses selection, crossover and mutation rules to create the next generation from the current population. Selection rules select the parents that contribute to the population at the next generation. Crossover rules combine two parents and mutation rules apply random changes to individual parents to form children for the next generation [5] [10].

As a selection algorithm, we started with Elite selection method [8] which gave better results for this problem case compared to other applied methods. Elite selection is a selection strategy where a limited number of individuals with best fitness values are chosen to pass to the next generation directly, without applying crossover and mutation operators. This way, good genes are avoided to be randomly destructed by crossover or mutation operators. In our implementation, number of elite individuals changes dynamically at each iteration by choosing a random ratio between 10% and 90%.

Number of iterations should not be so low or so high. If it is low, then the final result would probably not approach to the best result. If it is high, the result might be the best solution but it takes too much time to find it. Since time complexity is important and we do not want to overwhelm the gateway with a high processing load, we need to avoid large number of iterations. So in our case, 200 iterations gave us good results and our population size is limited with 40 and uses greedy cross-over. Mutation is performed by exchanging randomly chosen two genes.

When we apply genetic algorithm with Elite selection for the test case, the best chromosome found to be inserted into question marks of Table IV is as follows:

4 – 8 – 15 – 12 – 10 – 11 – 19 – 13 – 1 – 17 – 0 – 16 – 2 – 3 – 6 – 7 – 9 – 18 – 14 – 5

Therefore, for instance staff 1 will take care of 4th order, staff 2 will take care of 8th order and so on so forth. Total cost for this assignment is **77,5**.

In our tests, we also executed rulette selection method. Roulette selection is performed based on a roulette wheel with multiple parts, such that some of them are wider than the others. A proportion of the wheel is assigned to each of the possible selections based on their fitness value [7], wider parts for good fitness value, and smaller parts for inefficient ones. We use same iteration number (200) and population size (40). We executed the model repeatedly and we get an average result of approximately **149** which is far away from the solution found by the Elite selection. Since, Elite selection always gave good results, we used it as a basis in the rest of our study.

B. Extended Methodology

As well as the orders can be queued, we can also manage other various kinds of tasks. In the same way, specific staffs excluding particular ones accomplish specific tasks. For example, heavy box carriage from cold storage rooms must not be done by female staffs. Thus, female staffs are scored low (meaning high-cost) for these kind of jobs and they will not be assigned any. All the assignments are evaluated in gateway with a high-frequency. Every time a new task arrives at queue, gateway recalculates the assignments, as it is a very light-weight process and done in milliseconds.

To extend the Elite selection method used in genetic algorithm, we support it with a novel method which we name "Bet Prediction Selection" inspired from football leagues. This methodology aims to narrow the population for Elite selection to an efficient and qualified chromosome set. Elite selection method requires a population set as every selection method does. The population set size determines the complexity and completion time of genetic algorithm. Very narrow population size can provide a quick completion with a bad result, very wide one can cause a long completion time with a good result. Every population is generated randomly and nearly none of them is the optimal chromosomes. Besides being not optimal, many of them are very far far away from being optimal. Thus, any crossovers and mutation must be applied to a huge set.

Here "Bet Prediction Selection" methodology will eliminate the population into more qualified, efficient set. Moreover during the process time, this methodology can be applied to mutated, cross-overed population set at desired time. We will briefly use this method to eliminate the weakest links from the population set.

Now, let us describe details of "Bet Prediction Selection" with an analogy. Assume a football league with a group. So many people bets on matches, events, occurring, rankings in this league within a season by predicting whom to win. Bet companies determine the rates for teams assuming the preceding match's or event's result and apply some pre-defined rules. For example, B team has won 10 matches so far within 13 matches in season, and C team has won only 3 matches within same matches in the same season. If a match happen to occur between two teams, bet companies determine some rates for that match like B team will win the match with a 70% possibility, where C team will win the match with 10% possibility, and the rest for a draw. Also B team most likely be the champion in this season like 50%, where C team is 1%. So everybody bets their money on the teams, and if they predict true they return their money by multiplying it.

Here every chromosome in a population is actually a team. All chromosomes have some matches between each other. This methodology is somewhat similar to "Tournament Selection" in genetic algorithm. Every chromosome actually can be assumed as a season of a team. Every gene included in chromosome also can be assumed as matches. In the same way with Tournament Selection, the more teams included within a league, the higher the resulting selection pressure [9].

To instantiate it, let two chromosomes be like: 3-5-2-1-4-6-8-7-9-10 and 1-2-3-4-6-7-10-8-9-5. When they match each other, their genes given into fitness function one by one from the very beginning. For the first chromosome, first gene (match) is 3, for the second is 1. These values input into fitness function, and the cost is lower for gene 3 (first team). Thus, for the first match, the winner is the first team and a score goes for that team, in addition to this, cost difference are also scored and so on so forth… In these chromosomes, there are 10 of genes (matches). We define a pre-set number to play matches, let's say 5. So we do not play all of 10 games. We just play 5 matches, and after the matches, we look for the total points and cost differences, this can be assumed as a ranking table in football league. Here, betting logic will show up. According to points and cost differences, some bet rates will be defined for teams. Such as B Team wins the cup with 70%, where C Team does for 15%. These betting rates depend on the algorithm designer. We just applied basics. If a team wins 3 of 5 and loses 2 of 5, then the team will be rated as 60% so on so forth. For the total cost difference, the team who has the advantage will be additionally rated extra 20% (or another pre-set value or calculated value) win rate, while the other team will have the penalty for same rate.

We do not need to play all the matches, meaning that we do not need to iterate all the genes in the chromosome in order to decrease the time complexity. The number of matches to be played depends on us. If we increase that value, we predict the winners accurately, but it will have impacts on performance. If we decrease that value, we have good performance, but predicting winners may fail. Winners are a good portion of selection of chromosomes, which have better results than losers do. Starting with good quality chromosomes in population will give the better results within a short time. This is also what Elite Selection do. It always selects the elite chromosome near poor ones. A made-up scenario as in Table V would be better to explain what we are talking about.

TABLE V. EXAMPLE MADE-UP SCENARIO

Team No	Team Chromosome	Cost	Win/Lose	Win Rate	Bet Rates
1	1-2-3-4-5-6-7-8-9-0	63	3/1	80%	1,5
2	2-1-4-3-6-5-8-9-7-0	75	1/3	0%	5,25
3	1-3-5-6-7-8-9-0-2-4	15	5/0	120%	1,1
4	0-9-8-7-6-5-4-3-2-1	85	0/5	-20%	9,50
5	0-9-8-7-6-1-2-3-4-5	85	2/2	20%	2,75
6	2-4-6-8-0-1-3-5-9-7	75	2/2	60%	1,5

As in the example, we have chromosomes with 10 genes. We have decided to play 5 matches corresponding to the first 5 genes of chromosome. All the costs are made-up values, also bet rates are like real football game bet rates, and all of them are also made-up values related with their win percentage. In this league, we will select only the possible winning teams by decreasing team number from six to three. We can also get a risk factor, not being at the safe side. We can select the teams, which are more likely to lose. We do not know how the last 5 matches will end. Maybe the teams with low win rate will win eventually. So we can add a risk factor, so that we can select these teams by obeying this risk factor. It is just an option, but in this study we did not use this risk factor.

Now, we have narrowed the population set with chromosomes that are more qualified. Now, we can run elite selection method on this set.

C. Results

As an experimental setup, we have created two scenarios. In the first test scenario, there exists 20 Order/Tasks in job management screen. Product and delivery types of these orders are generated randomly. We apply Elite selection method and Elite Selection supported with Bet Prediction Selection (BPS) method for various population sizes. Initially we set number of iterations to 100. Results are illustrated in Table VI. The best result (BR) is found by using a high amount of population (more than 1000) with high number of iterations (more than 10000). For this setup the minimum cost (BR) is found as **104,75**.

TABLE VI. TEST RUN RESULTS

M	Pop	S.Pop.	PSU	It.	Res.	Time	SR
E	100	-	-	100	107,35	155	**97,51**
E	50	-	-	100	112,5	129	**92,60**
E	30	-	-	100	112,7	125	**92,41**
E+B	100	50	10	100	106,10	125	**98,71**
E+B	100	30	10	100	106,85	119	**97,99**

In the Table VI, column labels explained as follows: **M**: Method, **Pop**: Population, **S.Pop**: Shrinked Population, **PSU**: Play Season Until, **It**: Iteration, **Res**: Result, **Time**: Execution. Methodology E corresponds to only Elite Selection. E+B corresponds to BPS supported Elite Selection. While applying BPS, for 20 gene chromosomes, only 10 of them are matched with each other, which is represented as **PSU**. We have run lots of test runs, but table only represents the average results. Comparing population 50 over only Elite selection is slower and nearly 6% less accurate than the scenario supported with BPS with population shrinked from 100 to 50. Similar result also holds for population size of 30.

In the second test scenario, we increase the number of orders/tasks to 1000. Now we have a chromosome containing 1000 genes. Test results are shown in Table VII. Performance difference between E and E+B can be seen more clearly for this test scenario.

As it can be observed from Table VII, if we keep the chromosome longer (containing too much genes), our selection method creates a remarkable difference in terms of both time and result. For this scenario, best result is obtained as **6737,75** which is again the minimum result obtained within an environment containing a very high number of iterations and very rich population in a long time.

TABLE VII. TEST RUN RESULTS FOR 1000 GENES OF CHROMOSOMES

M	Pop	S. Pop.	PSU	It.	Res.	Time	SR
E	100	-	-	100	8228,2	28640	**77,88**
E	50	-	-	100	8405,2	13903	**75,25**
E	30	-	-	100	8504,7	8322	**73,78**
E	30	-	-	500	7894,7	48043	**82,83**
E	30	-	-	1K	7464,7	109923	**89,21**
E+B	1000	100	150	100	8043,2	26263	**80,62**
E+B	1000	50	150	100	8249,5	13026	**77,56**
E+B	1000	30	150	100	8376,2	8091	**75,68**
E+B	1000	30	500	100	8291,7	8702	**76,94**
E+B	1000	30	500	500	7709,7	46131	**85,57**
E+B	1000	30	500	1K	7214,2	105004	**92,93**

We also test the effect of the iteration number for both methods. We consider E with population size 30 and E+B with population size 30 shrinked from 100 for a chromosome set of 100 genes. We observe that nearly 1000 iterations would be enough for Elite Selection with BPS to converge to the best result, while nearly 1500 iterations are required for Elite selection without BPS. Also the results for different number of iterations are shown for 1000 genes in Table VII, and it is observed how E+B outperforms E with the same number of iterations.

D. Extending Bet Prediction Selection

In this paper BPS is used as an initializer. We applied it before Elite selection and observe how it would be useful. The results are not major, but something and promising.

This selection method can be replaced as a main selection method. Within the population, there will be a football like league within chromosomes. But unlike the tournament selection, we will not process the whole chromosome into fitness function. Only a pre-defined part will be processed and bet rates will be extracted. This methodology will make difference for huge chromosomes as we do not need to process all of the genes. At the mid-season the teams with lower percentage rates will be relegated from population. So rest ones will be crossover or mutated. This step will be taken frequently within season. In the end, a champion will show up and satisfy the best solution. Implementation and evaluation of this method would be an interesting piece of future work.

V. Conclusion

In this study, we propose an IoT architecture for quick service restaurants for intelligent allocation of various tasks to staffs with different capabilities. For the efficient and fair assignment of the tasks, we propose a genetic algorithm with a novel selection method, namely Bet Prediction Selection, that is used for initial parent selection. We evaluate the costs for various parameters, and observe that the proposed method provides remarkable decrease in the cost when applied as an initialization step. Using IoT based smart staff allocator have many benefits such as avoiding waste of time and resources, and freeing many key staff to another important jobs. Additionally, it also enables having a full control and tracking on task assignment system.

As a future work, we will extend Bet Prediction Selection method as the main selection method, rather than using only as an initializing method. In other words, we will implement and evaluate what we mentioned in the last part of the previous section, which we think as a promising approach.

Acknowledgment

This work is supported by BAPKO under D-Type Project and Ata Technology Platforms R&D Center.

References

[1] "Fast Food Industry Analysis 2018 - Cost & Trends", Web: Franchisehelp.com. [Accessed: 5- Agu- 2018]

[2] K. Aytaç, Ö. Korçak, "IoT Edge Computing in Quick Service Restaurants", *16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2018.

[3] J. S. Dean, "Staff Scheduling by a Genetic Algorithm with a Two-Dimensional Chromosome Structure", *In Proc of the 7th Conference on the Practice and Theory of Automated Timetabling*, 2008.

[4] "ZigBee Specification FAQ", Web: zigbee.org. [Accessed 1-Sep-2018]

[5] J. McCall, "Genetic algorithms for modelling and optimisation", *Journal of Computational and Applied Mathematics*, Volume 184, No. 1, 1 December 2005, pp. 205-222.

[6] X. Cai, K.N. Li, "A genetic algorithm for scheduling staff of mixed skills under multi-criteria", *European Journal of Operational Research*, Vol. 125, 2000, pp. 359 -369.

[7] "Genetic Algorithms - Parent Selection", Web: tutorialspoint.com. [Accessed 3- Sep- 2018]

[8] S. Jayaprakasam , Sharul K. A. Rahim, and Chee Y. Leow, "A Pareto Elite Selection Genetic Algorithm for Random Antenna Array Beamforming with Low Sidelobe Level", *Progress In Electromagnetics Research B*, Vol. 51, 2013, pp. 407–425.

[9] B. L. Miller, D. E . Goldberg, "Genetic Algorithms, Tournament Selection, and the Effects of Noise", *Complex Systems*, Vol. 9, 1995, pp. 193- 212.

[10] D. Beasley, D.R. Bull, and R.R.Martin, "An overview of genetic algorithms: Part 1, fundamentals", *University Computing*, Vol. 15, No. 2, 1993, pp. 56-69.

[11] L. Özbakir, A. Baykasoğlu, P. Tapkan, "Bees algorithm for generalized assignment problem", *Applied Mathematics and Computation*, Vol. 215, 2010, pp. 3782–3795.