

# A Genetic Type-2 Fuzzy Logic Based Approach for the Optimal Allocation of Mobile Field Engineers to their Working Areas

Andrew Starkey, Hani Hagrass  
The Computational Intelligence Centre, School of  
Computer Science and Electronic Engineering  
University of Essex, Colchester, UK

Sid Shakya, Gilbert Owusu  
Business Modelling and Operational Transformation  
Practice  
British Telecom, Adastral Park, Ipswich, UK

**Abstract**— In utility based service industries with a large mobile workforce, there is a need to optimize the process of allocating engineers to tasks (i.e. fixing faults, installing new services, such as internet connections, gas or electricity etc.). Part of the process of optimizing the resource allocation to tasks involves finding the optimum area for an engineer to operate within, which we term as work area optimization. Work area optimization in large businesses can have a noticeable impact on business costs, revenues and customer satisfaction. However when attempting to optimize the workforce in real world scenarios, mostly single objective optimization algorithms are used while employing crisp logic. Nevertheless, there are many objectives that need to be satisfied and hence multi-objective based optimization will be more suitable. Even where multi-objective optimization is employed, the involved systems fail to recognize that these real world problems are full of uncertainties. Type-2 fuzzy logic systems can handle the high level of uncertainties associated with the dynamic and changing environments, such as those presented with real world scheduling problems.

This paper presents a novel multi-objective genetic type-2 Fuzzy Logic based System for the optimal allocation of mobile workforces to their working areas. The method has been applied in a real world service industry workforce environment. The results show strong improvements when the proposed multi-objective type-2 fuzzy genetic based optimization system was applied to the work area optimization problem as compared to the heuristic or type-1 single objective optimization of the work area. Such optimization improvements of the working areas will result in improving the utilization of the workforce.

**Keywords**— Type-2 fuzzy logic, multi-objective genetic algorithms, work area optimization

## I. INTRODUCTION

For a company with a large mobile workforce, efficiency can have a significant impact on the operation costs and revenue. One aspect of efficiency that is key is the optimization of the areas engineers are assigned to. These areas (known as Working Areas (WAs) or Work Locations (WLs)) create the boundaries in which groups of engineers (teams) work within. These boundaries contain certain geographical areas and generate demand (tasks) for the resources.

Typical mobile workforces that provide utility services complete tasks such as installing service connections to households and to commercial premises, as well as fixing faults and general maintenance work. These types of tasks are linked to a specific geography which consists of customer premises as well as Service Distribution Points (SDPs). Grouping the SDPs together forms the mentioned Working Areas. There can be hundreds or even thousands of these SDPs, so a brute force or exhaustive search method will not find suitable solutions in a practical amount of time. These problems are known as Combinatorial Optimization (CO) problems [1]. Algorithms designed to tackle CO problems usually aim for a metaheuristic approach [2] because the optimization has to be completed within a reasonable amount of time as the environment changes on a daily basis. A common approach to tackling these large scale and complex optimization problems is Genetic Algorithms (GA) [3],[4],[5]. When using a GA there needs to be a way of testing how effective the created solution is at solving the problem. A good way to do this is to run the solution through a simulation. For WA optimization this would be a simulation of how effectively tasks would be completed given any setup of WAs. This would require calculating the paths engineers would take to complete tasks, so that their estimated travel distance and time can be calculated. This essentially links into the Travelling Salesman Problem (TSP). The goal of the TSP is for a salesman to visit all cities in a given set only once and end up at the starting city. This has to be done in the shortest distance (minimum cost). However the number of potential paths increases exponentially with the increase in the number of cities the salesman has to visit [6].

Given the complexity and multiple objectives of these large scale optimization problems, traditional single objective genetic algorithms may not be appropriate. This is because they fail to take into account the conflicting nature some of the optimization objectives may have. One way of solving this problem is to use a Multi-Objective Genetic Algorithm (MOGA). Another area that aims to improve the solutions generated for the WA optimization problem is fuzzy logic.

The reason fuzzy logic should be applied to WA optimization is the potentially high levels of randomness and uncertainty which face the problem of WA optimization in changing and dynamic environments, which include:

- Uncertainties in the data used for optimization as the data used is collected or estimated from real world data.
  - Uncertainties on the available skills per day due to engineers falling sick or going on holiday.
  - Uncertainties affecting the travel times and distance.
- As the travel times between locations are given by a route planner which may not reflect road works, traffic collisions, toll roads or rush-hour traffic etc.
- Uncertainties affecting the estimated job completion times. The average time to complete a job, of a particular type, in each SDP is used to estimate the job completion time. However each engineer has their own rate of efficiency that is not used.

There are a number of examples where real world problems use genetic algorithms to solve the issues presented [7],[8],[9]. There are also examples of multi-objective GAs being used to solve real world problems [10],[11]. However none of the existing solutions which employed single or multi objective GA for real world WA optimization employed fuzzy logic.

The introduction of fuzzy logic can potentially increase the accuracy of the optimization because fuzzy logic can be used to account for the real world uncertainties. However the type-1 Fuzzy Logic Systems (FLSs) cannot fully handle the high level of uncertainties associated with the real world dynamic and changing environments as the type-1 FLS employ the crisp and precise type-1 fuzzy sets. Type-2 FLSs can handle high uncertainty levels as they employ type-2 fuzzy sets which provide through their Footprint of Uncertainty (FOU) and third dimension additional degrees of freedom to enable handling higher uncertainty levels. There have been notable examples of type-2 FLS outperforming the results of type-1 fuzzy logic system [12],[13], [14].

This paper presents a novel multi-objective genetic type-2 Fuzzy Logic based System for the optimal allocation of mobile field engineers to their working areas.

Section II will present an overview on the problem description of WA optimization. Section III will provide high level overview on type-2 FLSs. Section IV will present a high level overview on the employed multi-objective genetic algorithm, NSGA-II. Section V will present the proposed multi-objective genetic type-2 Fuzzy Logic based System for mobile field engineer area optimization. Section VI will present the conclusions and future work.

## II. OVERVIEW OF THE WORK AREA OPTIMIZATION PROBLEM

### A. Overview on Work Areas

A typical service industry with a large field workforce usually operate within a fixed geographical structure, i.e. the country is divided into different regions and sub regions. This

helps to effectively manage the field workforce and focus the allocation of work. At the lowest level of this structure, the areas are known as Working Areas (WAs) which are made up of a collection of Service Distribution Points (SDPs). The SDPs serve domestic and commercial properties by connecting these properties to services such as electricity, gas, water or telecoms depending on the service provided by the workforce. Fig 1a illustrates how the UK might be subdivided into regions and sub regions. Fig 1b shows a sub region, which is divided into 5 WAs, with the groups of SDPs for each.

Optimization of WAs involves grouping or regrouping these SDPs based on the latest demand and capacity information, such that the effectiveness and efficiency of service is maximized. Note that high density areas are smaller than rural areas. This is because the WAs need to be balanced in terms of work, where high density urban areas provide more work per SDP than low density rural areas. The goal is to have the engineers service as many tasks as possible at the lowest cost

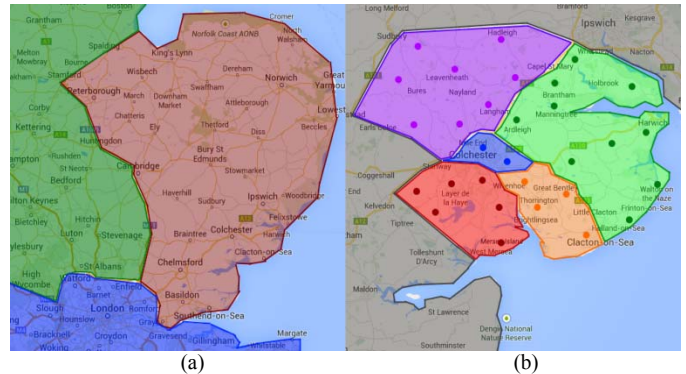


Fig. 1. a) Regional Areas. b) WAs within a Sub Region.

### B. Objectives and Constraints

The WA optimization process has a number of objectives which need to be satisfied. They are as follows:

- **Maximize Coverage:** Coverage is the amount of tasks that are estimated to be completed.
- **Minimize Travel:** Minimizing traveling distance increases the amount of available time for each engineer and also decreases costs. However minimizing travel directly conflicts with maximizing coverage. This is because an engineer (in the majority of cases) will be required to travel to each task. As coverage increases, travel also increases.
- **Maximize Utilization:** Utilization is the percentage of time an engineer is completing tasks. Unutilized time is when the engineer is idle or travelling.
- **Area Balancing:** WAs should be evenly balanced with the amount of work they contain. This links back to smaller WAs for urban areas and larger WAs for rural areas. This is a minimization objective. The difference between the biggest area and smallest area, in hours worth of work, is calculated. The smaller this value the better.

There are a number of constraints that need to be looked at and included in the optimization. All of the engineers won't be

working at all times (as some of them might fall sick, have holidays or day offs), so there is a degree of workforce shrinkage that needs to be taken into account. Of the engineers that remain, they can only be assigned tasks that they are qualified to complete. Of these tasks, each engineer has preferred tasks that they work on. Taking this into account can help improve the average time taken to complete tasks.

Another constraint is that each engineer is limited by the amount of work they can do each day (travel time has to be included in this). In addition, each team has to be equal in size and WAs should not cross large rivers.

### III. OVERVIEW ON TYPE-2 FUZZY LOGIC SYSTEMS

Fuzzy Logic Systems (FLSs) have been credited with providing white box transparent models which can handle the uncertainty and imprecision. However, the vast majority of the FLSs were based on type-1 fuzzy logic systems which cannot fully handle or accommodate for the uncertainties associated with changing and dynamic environments. Type-1 fuzzy sets handles the uncertainties associated with the FLS inputs and outputs by using precise and crisp membership functions [15]. Once the type-1 membership functions have been chosen, all the uncertainty disappears, because type-1 membership functions are totally precise [12], [15]. Due to the change in the individual engineer circumstances and the uncertainties present in the surrounding environments, the chosen type-1 fuzzy sets might not be appropriate anymore. This can cause degradation in the FLS performance and we might end up wasting time in frequently redesigning or tuning the type-1 FLS so that it can deal with the various uncertainties faced. Type-2 FLSs which employ type-2 fuzzy sets can handle such high levels of uncertainties to give very good performances.

A type-2 fuzzy set is characterized by a fuzzy membership function, i.e. the membership value (or membership grade) for each element of this set is a fuzzy set in  $[0,1]$ , unlike a type-1 fuzzy set where the membership grade is a crisp number in  $[0,1]$  [15]. The membership functions of type-2 fuzzy sets are three dimensional and include a Footprint of Uncertainty (FOU), it is the new third-dimension of type-2 fuzzy sets and the footprint of uncertainty that provide additional degrees of freedom that make it possible to directly model and handle uncertainties [12], [15].

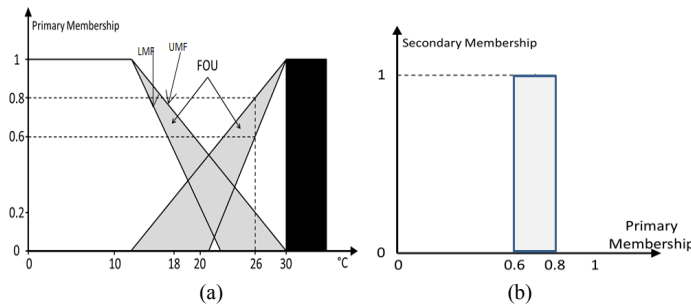


Fig. 2. a) An interval type-2 fuzzy set- primary membership function. b) An interval type-2 fuzzy set secondary MF at a specific point  $x$ .

As shown in Fig 2a, the Interval Type-2 (IT2) fuzzy set  $\tilde{A}$  can be represented in terms of the Upper Membership

Function (UMF) (denoted by  $\bar{\mu}_{\tilde{A}}(x), \forall x \in X$ ) and the Lower Membership Function (LMF) (denoted by  $\underline{\mu}_{\tilde{A}}(x), \forall x \in X$ ) as follows:

$$\tilde{A} = \int_{x \in X} \left[ \int_{u \in [\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)]} 1/u \right] / x \quad (1)$$

The UMF and LMF are bounds for the  $FOU(\tilde{A})$  of an IT2 fuzzy set  $\tilde{A}$ . As shown in Fig 2b, in an IT2 fuzzy set the secondary membership function is equal to 1 for all the points in the primary membership for  $\forall x \in X$ .

In type-2 FLSs, the crisp inputs are fuzzified to input type-2 fuzzy sets which are fed to the inference engine which maps the input Type-2 Fuzzy sets to output Type-2 fuzzy sets using the rule base. The output set is then processed by the type-reducer in the type reduction section which generates a type-1 output set.

### IV. OVERVIEW ON NSGA- II MOGA

Genetic algorithms (GAs) are based on the theory of survival of the fittest, where solutions are individuals in a population and these individuals are assessed to see how well they satisfy the current objective(s). The value given by this assessment is known as the fitness value. In single objective GAs, if there are more than one objective, then the objective values need to be combined into a single objective function. If the objectives are complementary or do not have any correlation then the objective function may be sufficient. In cases where the objectives conflict then the use of a fitness function starts to show its weakness. Some examples of conflicting objectives include:

- Minimizing Cost while Maximizing production
- Minimizing CO2 emissions while Maximizing Transport Capacity
- Maximizing Customer Satisfaction while Minimizing Staff

The problem with these conflicting objectives is that they cannot both be 100% satisfied without causing significant damage to the other objective. For example we can easily minimize costs to 0, however production would also be 0 so this situation is not acceptable in real world problems.

Multi-Objective Genetic Algorithms (MOGAs) compare the results of each objective between solutions. This means that the value for the first objective for one solution can be compared to the value for the first objective of another solution explicitly. In this way it's clear which solution is stronger in this objective. This is instead of having all values amalgamated into one fitness value and not really knowing how the objective values compare.

Given that each objective is compared between solutions there needs to be a method of deciding if one solution is better than another. One way of doing this is by 'Domination' as is done in the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [16]. Domination determines if one solution

dominates another by setting out conditions. These conditions are as follows (to determine if solution A dominates solution B):

- Solution A has no objective value that is worse than the respective objective value in B.
- Solution A has at least one objective value that is better than the respective objective value in B.

If both of these conditions are met it would be determined that A dominates B, meaning solution A is the better solution. If each solution is compared with every other solution in the population in the same way, the domination count can be calculated. The domination count is the number of solutions that dominate the current solution.

Once the domination count has been calculated a simple sorting algorithm can be used to order the solutions from best to worst. The solutions with a domination count of 0 (no solutions are deemed better) are grouped together to form the Pareto front. This is the set of solutions that are all deemed to be the best and selecting any of these solutions will be the most suitable for the problem. All other fronts (sets) are made up of the other solutions and are grouped based on their domination count.

### V. THE PROPOSED MULTI-OBJECTIVE GENETIC TYPE-2 FUZZY LOGIC BASED SYSTEM FOR MOBILE FIELD ENGINEER WA OPTIMIZATION

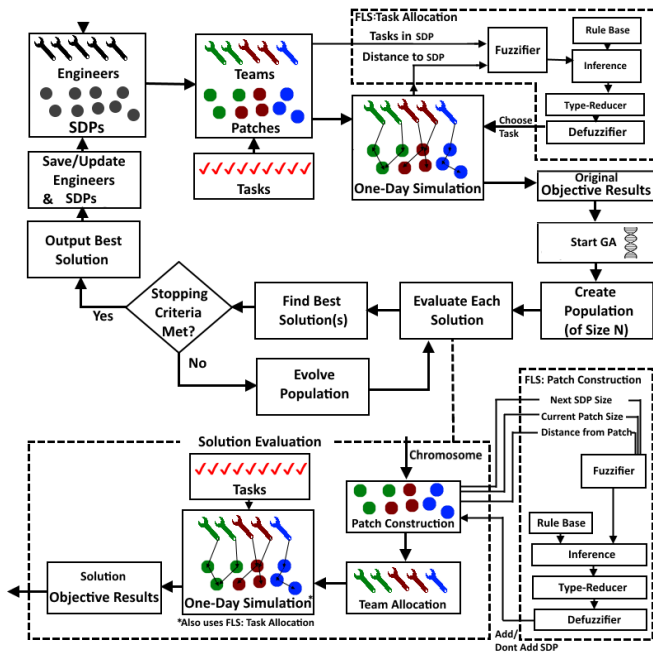


Fig. 3. The Proposed Multi-Objective Genetic Type-2 Fuzzy Logic Based System for Mobile Field Engineer WA Optimization

Fig.3 shows an overview of the proposed multi-objective genetic type-2 fuzzy logic based system for mobile field engineer WA optimization, where the first step is to collect the list of engineers and the list of SDPs to optimize. The

engineers and SDPs will already be grouped together into teams and WAs (Also known as patches).

The system, then retrieves the current setup of patches with their respective teams. This configuration is then put through the one-day simulation to assess how the current setup is performing. The one-day simulation cycles through each engineer and assigns them tasks based on their skills and the patch they are in. The simulation will attempt to assign the closest tasks to the engineer. Once a task has been assigned it will be removed from the task list.

Each engineer will be assigned tasks until their time has been filled. Each engineer is allocated 7 hours (the average for a working day). When an engineer is assigned a task, the time will be added to the utilized time of the engineer, while the time it takes to travel to the task will be added to the engineers travel time (part of the engineers unutilized time).

The distance traveled is also stored per engineer. The simulation will also stop assigning tasks if there are no more available tasks for that engineer to complete in their patch. Any remaining time will be added to the engineer's unutilized time.

The one-day simulation is a stage where the type-2 FLS for task allocation can be applied. When choosing which job to assign to an engineer, the type-2 FLS is supplied with two inputs which are: the distance and time to the task and the number of tasks at the SDP where the task is located. This helps the simulation take into account the uncertainty of the travel time and to direct the engineer to SDPs with more tasks. More on this can be found in section V.A.

Once each engineer has been cycled through, the system will calculate the objective results. The first objective to calculate is the coverage. This is the total amount of hours of completed work. This is calculated by summing all the utilized time of the engineers. The second objective is the total travel distance. This is calculated by summing all the total travel distances of the engineers. The third objective is utilization, this is calculated by dividing the utilized time of an engineer by the max time (7 hours). This objective is then expressed as an average across all engineers. The final objective is balancing which adds up all the task time per patch and finds the number of hours different between the largest patch and smallest patch. Ideally this difference value should be 0, meaning perfect balancing.

Given that the current setup has been evaluated, these values can be used as a rough benchmark for the optimization process to improve upon. The system gives the user the option to adjust any of the GA's parameters before the optimization process is started. For the experiments outlined in VI of this paper we kept the parameters for the GA constant across all experiments. The GA utilized a tournament selection for the crossover operator and a one bit mutation operator.

When the GA is running each of the solutions it generates need to be evaluated. The first step to this is building the patch setup from the centre SDPs (held by the genes of the solutions). SDPs in the same patch cannot be separated. This is either by rivers or by other patches. The patch construction works in the following way. Each center point works out who



its neighbouring SDPs are. Then out of these neighbours, the system works out which is the closest. If no other patch has deemed that SDP to be the closest, it will be added to the patch. The next patch will do the same. Each time an SDP is removed from the list and added to a patch, each patch has to recalculate who its available neighbours are.

Patch construction is a step where the type-2 FLS for patch construction is applied. This helps to decide if an SDP should be added to a patch. The list of all available SDPs will be passed to the type-2 FLS in Fig.3 whose inputs are the size of the SDP (in hours), the size of the patch (in hours) and the distance to the SDP (in km). More on this can be found in section V.B.

Once the patches have been constructed the teams for each patch need to be assigned. This first step in this process is to assign each engineer to the patch they live in (or are closest to, if they do not live in any patch). This will usually mean the teams are extremely unbalanced as city/town patches will have over populated teams and rural patches will have underpopulated teams. The next step is to balance out the teams. This is done by a bidding process. The system will cycle though each overpopulated patch and ‘sell off’ its engineers to the highest bidders. Each underpopulated patch will cycle through the current overpopulated patch’s engineers and give it a bid value. If there are no other bids for this engineer they will move over to the underpopulated patch, if there are other bids the highest bid wins. The bid value is made up of the distance the engineer is from the underpopulated patch, how much their skills are needed and the level of under-population the patch is at. Once the bidding process is complete the engineers should be spread as best as possible between the patches. The newly constructed patches and teams will then go through the same one-day simulation process as the original setup. The objective values for this solution will be calculated. The GA will carry out the ‘Solution Evaluation’ for every solution it generates.

With each solution in the population evaluated, regular GA processes are resumed. The stopping criteria that is currently being utilized in the system is the maximum number of generations. Once the GA has stopped the results are reported and output files can be generated. These output files list each engineer and their newly assigned patch and the structure of the new patches.

#### A. The Type-2 FLS for Task Allocation

Figs 4 and 5 show the interval type-2 fuzzy sets representing the two inputs of the type-2 FLS for task allocation which are the distance to task and the tasks in the SDP respectively. Fig.6 shows the interval type-2 fuzzy sets representing the output of the type-2 FLS which is probability of picking up a task. The average distance to a task (AD in Fig. 4) is calculated for the area being optimized. This is done before the initial one-day simulation when the teams and SDPs are first loaded. The average amount of work time in an SDP for the area (AW in Fig. 5) is also calculated at this point

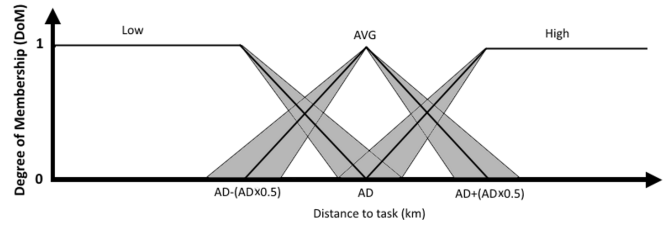


Fig. 4. Type-2 fuzzy sets representing the Distance to task input.

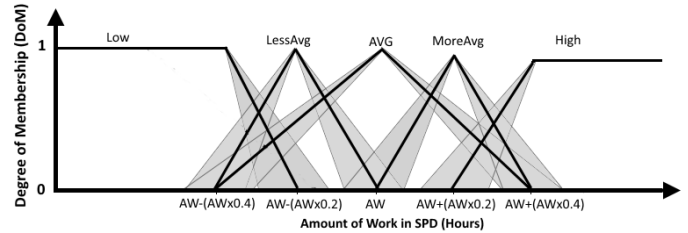


Fig. 5. Type-2 fuzzy representing the Tasks in SDP input.

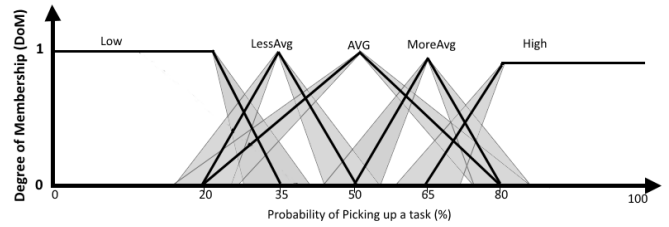


Fig. 6. Type-2 fuzzy sets representing the probability of choosing task output.

The percentage of uncertainty (which determines the FOU endpoints) is a variable in the experiments where the base points of the membership functions were tuned by running experiments to find the most suitable setup. The values for the average distance (AD) and average amount of work (AW) had to be calculated so that their values were relative to the area that was being optimized.

For example an average distance per job in London might be 100m but in the Scottish Highlands this value might be 5km or more. Having the base points relative to the area is important, else input values will be wrongly categorized relative to the local area. Table I shows a sample list of the rules used in this type-2 FLS. All the type-2 FLSs used in this paper employ the centre of sets type-reduction.

TABLE I. SAMPLE RULES FOR TASK ALLOCATION

Distance To Task	Tasks at SDP	Consequence
If Distance is Low	and Tasks are Low	then Probability is Average
If Distance is Low	and Tasks are High	then Probability is High
If Distance is Average	and Tasks are High	then Probability is MoreAvg
If Distance is High	and Tasks are Low	then Probability is Low
If Distance is High	and Tasks are High	then Probability is Average

### B. The Type-2 FLS for Patch Construction

Figs 7, 8 and 9 show the interval type-2 fuzzy sets representing the three inputs of the type-2 FLS for patch construction. These sets are the patch size average, SDP size average and average distance respectively. Fig.10 shows the type-1 fuzzy sets representing the output of the type-1 FLS which is the chance of an SDP being added. When the area to be optimized is initially loaded up the average patch size (in hours of work) is calculated along with the average SDP size. This is because these values can vary a lot between urban and rural areas. The base points of the membership functions were tested to see if reasonable categorization of SDP and patch sizes were given. The task of this FLS is to more sensibly add SDPs to patches. The center points of the patches are provided to the fuzzy system (these center points are the initial SDPs allocated to each patch). The size of the patch is re-calculated each time an SDP is added to it. When deciding if an SDP is to be added to a patch it will go through the Add/Not Add outputs whose fuzzy set are shown in Fig. 10. The Add/Not Add membership functions were designed in such a way that a rule with a not add consequence would have more of an impact on the final outcome than an add consequence. The output values are compared between the patches, with the SDP being added to the patch with the highest output value.

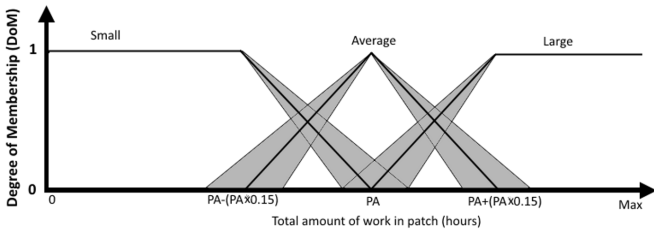


Fig. 7. The type-2 fuzzy sets representing the patch aize average input.

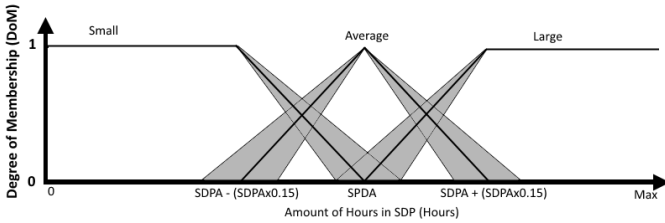


Fig. 8. The type-2 fuzzy sets representing the SDP size average input.

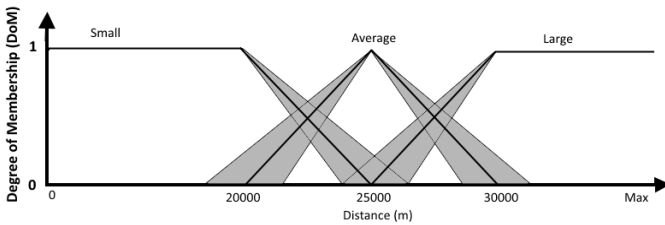


Fig. 9. The type-2 fuzzy sets representing the average distance input.

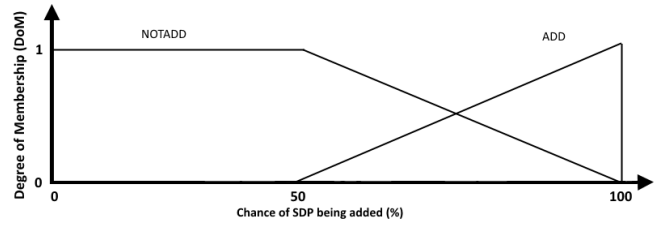


Fig. 10. The type-1 fuzzy set representing the Add/Not Add output.

Table II shows a sample list of the rules used in this type-2 FLS.

TABLE II. SAMPLE RULES FOR PATCH CONSTRUCTION

Patch Size	Distance to SDP	SDP Size	Consequence
If Patch is Small	and Distance is Small	and SDP is Small	then Add
If Patch is Small	and Distance is Small	and SDP is Large	then Add
If Patch is Large	and Distance is Large	and SDP is Large	then NotAdd
Average			
If Patch is Large	and Distance is Small	and SDP is Average	then NotAdd

## VI. EXPERIMENTS AND RESULTS

The aim of the experiments is to take an existing structure of WAs with its current patch set up and teams of engineers, then run it through the optimization process to see how well the WAs get optimized. These experiments are then repeated with potential improvements added to the optimization to see the impact these potential improvements will make. The real world tool created for this process is shown in Fig. 11. The tool allows the visualization of the working areas and SDPs.

The experiments involved altering the optimization process by gradually increasing the use of more advanced optimization methods. The process started by comparing the use of single and multi-objective GAs and then progressed to evaluate the effect of employing type-1 and type-2 FLSs

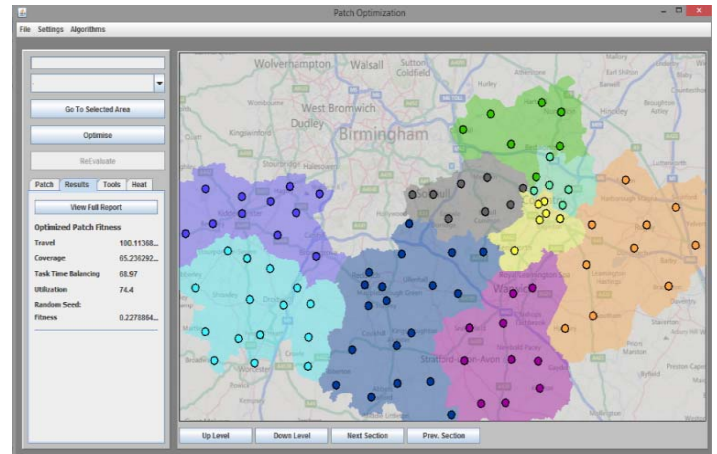


Fig. 11. The mobile field WA visualization & optimization tool.

### A. Progressive Results

Due to the limited space, in this section, we show a sample set of experiments which were performed in one sequential real time test, exactly how they would be run in the real world. Table III shows the results from the progressive tests. The original statistics obtained from the heuristics and crisp logic are given in row 1. The first step is to optimize this WA with the Single Objective GA (SOGA) and crisp logic. Row 2 shows that on this occasion the SOGA failed to optimize in any objective. This means that the optimization would have to be run again and the GA setting would need to be tuned for this specific area to get a better result. This would cause frustration to the user and cost time.

Row 3 shows us the most suitable solution from the MOGA and crisp logic. On this occasion the MOGA has optimized in Balancing and for Travel it is less than 1% worse, so that can be seen as the same. However the MOGA has failed to optimize on coverage and utilization. If the user was looking to only improve on Balancing and was happy to suffer the reduction in the other two objectives this may be acceptable, else the optimization would need to be run again.

Row 4 shows the most suitable solution from the MOGA using type-1 FLSs in the optimization. Here we can see that the MOGA has now optimized in two objectives. However Coverage and Utilization still suffer. But they suffer less than if the MOGA did not use the type-1 FLSs. There is an increase in both Coverage and Utilization over the MOGA that does not use any FLSs.

Finally row 5 shows the most suitable MOGA result with type-2 FLSs (that has been tuned to 1% uncertainty). On this occasion two objectives have been optimized and the remaining two do not suffer noticeably. This gives the user a solid result and can confidently say that this new patch is better than the old patch. This is on one run of the optimization and with no specific tuning of the GA required. Which is great from a user's point of view.

As a result we can say that these results support a multi-objective genetic type-2 fuzzy logic based system for mobile field engineer WA optimization. The system has given the best improvement from the original existing WAs in reducing travel by 60%, reducing on the un-balanced patches by 70% whilst having roughly the same coverage and the same utilization. The type-2 FLS based MOGA results are better than their type-1 MOGA results where the type-2 FLS reduced the travel by 1.5 % and decreased balancing by 51% and improved the coverage by 2.8% and the utilization by 2.3%.

TABLE III. PROGRESSIVE RESULTS

	Travel (km)	Coverage (hours)	Balancing (hours)	Utilization (%)
<b>Original</b>	<b>172.00</b>	<b>71.34%</b>	<b>68.96</b>	<b>63.88%</b>
<b>SOGA</b>	<b>187.16</b>	<b>68.86%</b>	<b>110.16</b>	<b>61.67%</b>
<b>MOGA</b>	<b>173.26</b>	<b>68.46%</b>	<b>54.21</b>	<b>61.30%</b>
<b>MOGA T1-FLS</b>	<b>67.01</b>	<b>69.68%</b>	<b>62.09</b>	<b>62.40%</b>
<b>MOGA T2-FLS</b>	<b>68.15</b>	<b>71.25%</b>	<b>30.08</b>	<b>63.81%</b>

### B. Subjective Results

Fig. 12, 13, 14 and 15 show the visualization of how the results change with each incremental improvement of the proposed system. Fig. 12 shows the SOGA try and divide the selected area, into 9 WAs. The selected area includes both rural and urban areas, including the densely populated city area and suburbs. The Single objective optimization has split the core city area (circled in Fig 12.) up into 3 WAs, this is not good as engineers will have to keep travelling in or out of the city. The other WAs are either too large or too small.

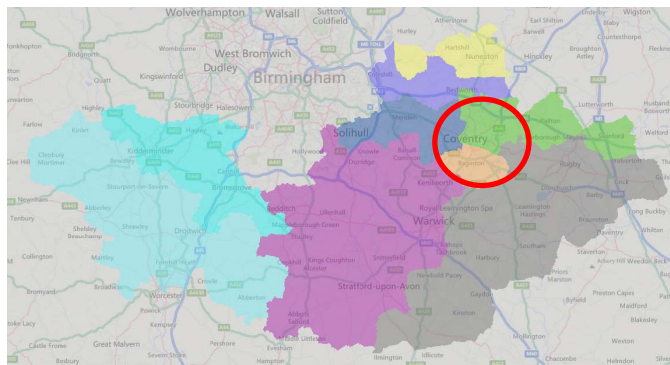


Fig. 12. SOGA Optimization Result (City Area Circled)

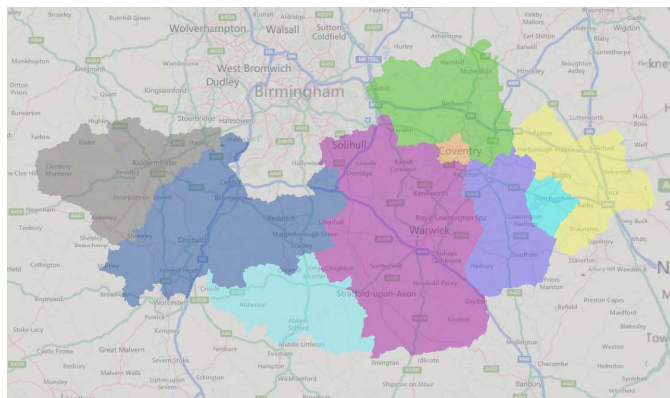


Fig. 13. MOGA Optimization Result

Fig. 13 shows one of the solutions on the Pareto front from the MOGA with crisp logic. This solution is slightly better as it has sectioned off the city center. But this WA is now too small as the outside of the city forms part of another WA to the North. This has also left one of the suburbs of the city in a very oversized WA. However the other WAs are of reasonable size.

Fig. 14 shows a solution that used the MOGA with type-1 FLSs in the optimization process. This has done a slightly better job of sectioning off the core city area, but there are a few SDPs that the WA has not included. There is also a WA that is too small, and the suburb WAs are still quite large.



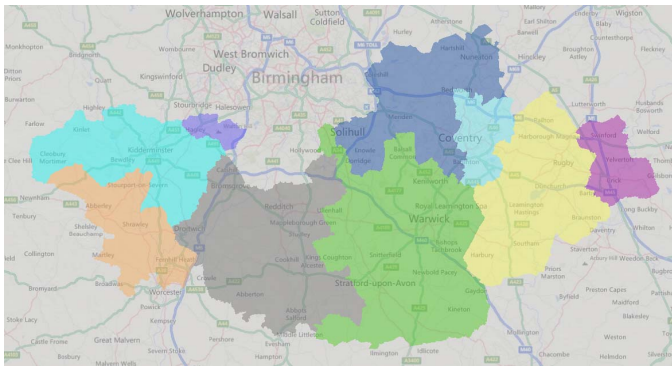


Fig. 14. MOGA with Type-1 FLS

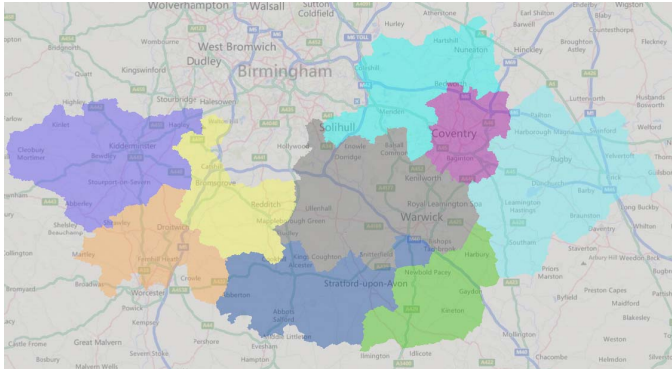


Fig. 15. MOGA with Type-2 FLS

Figure 15 shows a solution that has replaced the type-1 fuzzy with type-2 fuzzy logic in the MOGA. This solution has done a good job of sectioning off the core city area. Each WA is more balanced in size and even the large town to the west is its own WA. There also seems to be reasonable utilization of the road networks in the area. MOGA with Type-2 Fuzzy has produced the most sensible WA designs visually (this is important to the engineers and managers who have to accept these designs) as well as the best results from the simulation.

## VII. CONCLUSIONS & FUTURE WORK

In this paper we have presented a multi-objective genetic type-2 fuzzy logic based system for mobile field engineer WA optimization. To fully evaluate the proposed systems, we ran them through several experiments each designed to assess the impact of the different methodologies. The results from these experiments showed that a multi-objective system was able to optimize in more objectives than a single objective system. The results also showed that including type-1 fuzzy logic systems on the resource allocation and the patch construction parts of the optimization improved the results the system generated. Hence, the experiments showed that the MOGA based type-2 fuzzy system outperformed the counterparts that used type-1 FLS, crisp logic and SOGA. .

For our future work, we intend to carry out deployments in the field. We aim to optimize the type-2 FLS parameters via a fast converging optimization method. WA optimization is just one aspect to the overall vision of workforce optimization. Another domain that should be tackled is the optimization of

the engineers. Each engineer has a set of skills, however they can be trained to gain more skills or they can stop being assigned tasks they underperform in, effectively losing a skill. Combining this with the WA optimization will help to increase the overall benefits.

## VIII. REFERENCES

- [1] O.Turchyn, "Comparative Analysis of Metaheuristics Solving Combinatorial Optimization Problems", 9th International Conference on the Experience of Designing and Applications of CAD Systems in Microelectronics, Lviv-Polyana, Ukraine, 2007, pp. 276-277.
- [2] W. Fanm Z, Gurmu and E. Haile, "A Bi-Level Metaheuristic Approach to designing Optimal Bus Transit Route Network", 3rd Annual International Conference on Cyber Technology in Automation, Control and Intelligent Systems, Nanjing, China, 2013, pp. 308-313.
- [3] R. Domberger, L. Frey and T.Hanne, "Single and Multiobjective Optimization of the train staff planning problem using genetic algorithms", IEEE Congress on Evolutionary Computation, Hong Kong, 2008, pp. 970-977.
- [4] Y. Liu, S. Zhao, X. Du and S. Li, "Optimization of Resource Allocation in Construction Using Genetic Algorithms", Proceedings of the 2005 International Conference on Machine Learning, Guangzhou, China, 2005, pp. 18-21.
- [5] J. Tanomaru, "Staff Scheduling by a Genetic Algorithm with Heuristic Operators" International Conference on Evolutionary Computation, Perth, WA, Australia, 1995, pp. 456-461.
- [6] D.N. Mudaliar and N.K. Modi, "Unraveling Travelling Salesman Problem by genetic algorithm using M-Crossover Operator", 2013 International Conference on Signal Processing Image Processing & Pattern Recognition, Coimbatore, India, 2013, pp. 127-130.
- [7] T. Oda, A. Barolli, E. Spaho, L. Barolli, F. Xhafa and J. Iwashige, "Genetic Algorithms for Node Placement in WMNs: Effect of Changes in Population Size and Number of Generations", 7th International Conference on Broadband, Wireless Computing, Communication and Applications, Victoria, BC, Canada, 2012, pp. 356-361.
- [8] Y. Zhang, "Research on Human Resource Allocation Optimization Based on Genetic Algorithm from the Perspective of Two-way Choice Model", International Conference on Educational and Information Technology, Chongqing, China, 2010, pp. 380-383.
- [9] H. Zhaodong, C. Wenbing, Xiao Yiyong and L Rui, "Optimizing Human Resources Allocation on Aircraft Maintenance with Predefined Sequence" International Conference on Logistics Systems and Intelligent Management, Harbin, China, 2010, pp. 1018-1022.
- [10] X. Chen-Guang, Z Mai-huan, L. Yue-Peng and C. Nan-Xiang, "The Water Resource Optimal Allocation Based on Multi-Objective Genetic Algorithms", International Conference on Computational and Information Sciences, Chengdu, China, 2010, pp. 877-880.
- [11] J. Zhang, W. Wang, X. Xu and J. Jie, "A Multi-objective Particle Swarm Optimization for Dual-Resource Constrained Shop Scheduling with Resource Flexibility" IEEE Symposium on Computational Intelligence for Engineering Solutions, Singapore, 2013, pp. 16-19.
- [12] H. Hagrass, "A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots", IEEE Transactions on Fuzzy Systems, Vol. 12, No. 4, pp. 524-539, 2004.
- [13] C. Lynch, H. Hagrass and V. Callaghan, "Embedded Type-2 FLC for Real-Time Speed Control of Marine & Traction Diesel Engines" 14th IEEE International Conference on Fuzzy Systems. Reno, NV, USA, 2005, pp. 346-352.
- [14] H. Hagrass, M. Colly, V Callaghan, G. Clarke, H. Duman and Arran Holmes "A fuzzy incremental synchronous learning technique for embedded-agents learning and control in intelligent inhabited environments", Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, Hawaii, USA, 2002, pp. 139-145.
- [15] N. N. Kamik, J. M. Mendel, "Centroid of a Type-2 Fuzzy Set", Information Science Vol 131, pp. 195-220, 2001
- [16] K. Deb, A. Pratap, S Agarwal and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, pp. 182-197. 2002.