Routeing Strategy for Coverage Path Planning in Agricultural Monitoring Activity using UAV

A.F Hawary*, A.J Chipperfield

***Abstract*—** This article proposes a new combinatorial path planning method for Unmanned Aerial Vehicle (UAV) for agricultural monitoring using multi-objective optimisation approach. The purpose of the algorithm is to optimise two conflicting objectives in UAV operation, path length and coverage lost (CL). The method is constructed based on three planners namely Route Planner (RP), Path Planner (PP) and Coverage Planner (CP). The algorithm in each planner is developed using a modified version of Genetic Algorithm (GA) and Dijkstra Algorithm (A*) respectively. The solution for routeing problem is based on the problem of Travelling Salesman Problem (TSP) and obstacles avoidance. A new method of genetic mutation is proposed to improve the premature convergence as well as minimising the crossing path in the routeing process. A distance approximation function is used if the original trajectory is not optimised due to the obstacles. In addition, coverage estimation is proposed to evaluate the coverage loss within the search boundary *R*. From the result, the combined algorithm have produced different candidate solutions as the value of *R* varies hence, provide the operator the best potential solution.

***Keywords*—** Agricultural Monitoring, Genetic Algorithm, Multi -Objective Path planning, Unmanned Aerial vehicles.

I. Introduction

Due to the increasing demand and stringent requirement in agricultural products, an effective monitoring is required [2]. The biggest drawback of the ordinary method using ground vehicles is due to limited manueverity particularly in hard-to-reach areas as compared to aerial monitoring using UAV [1]. The use of UAV in agricultural sector such as crop monitoring, seed spreading and water spaying have proven to increase the productivity [3]. However, such activities require a proper path planning to ensure the UAV operates as desired.

A general problem of UAV path-planning is to determine how far the trajectory of the aircraft fulfils the user's requirement. Several path planning algorithms for UAVs have been proposed such as A* searching algorithm [4], Voronoi diagram[5], Dynamic Programming [6], the Rapidly-exploring Random Tree (RRT)[7] and so on. In addition to the standard single-objective path planning, several multi-objective path planning algorithms have also been proposed in [8], [9]. However, the papers do not emphasis the coverage in agricultural context. Recently, Ahmed [10] proposed a path planning that optimises the turning curve of the UAV trajectory to reduce the distance as well as the cost but coverage was not considered.

The reason why the coverage path planning is important is that, the agricultural practice in Malaysia (e.g., Palm oil, paddy field, corn field) are planted at different stage at different section of lands. Ordinary method using the method shown in Fig.1 below is no longer appropriate as it may not optimise the distance.

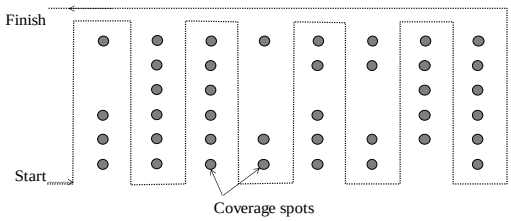

Fig.1: Standard agricultural layout and survey route using Boustrophedon method.

Fig.1 shows a standard agricultural layout using Boustrophedon method [11]. It literally means, writing of alternate lines in opposite directions (as from left to right and from right to left). Despite full coverage, this method is forced the trajectory to visit all the areas although the areas are not planted. Therefore, a more efficient method is proposed to reduce the coverage redundancies as illustrated in Fig.2. In addition, a method of obstacle's avoidance is also proposed.

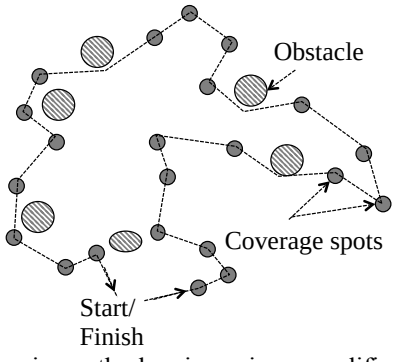

Fig.2: A route using path planning using a modified Travelling Salesman Problem.

Fig.2 consists of several coverage areas and few obstacles that closely represents the nature of the unstructured plantation. In this scenario, the UAV has to visit every spot and return to the finish point after all the spots are visited. The solution resembles the optimisation of a Travelling Salesman Problem (TSP)[12] with collision-free paths to guarantee the safety of

A.F Hawary* , A.J Chipperfield,, *School of Aerospace Engineering, Universiti Sains Malaysia, Nibong Tebal, Pulau Pinang, Malaysia.. Faculty of Engineering and the Environment, University of Southampton, Southampton, SO17 1BJ United Kingdom.

http://dx.doi.org/10.17758/EAP.EAP816005

the operations. As far as the optimisation is concerned, the route should minimize the flying distance and maximise the ground coverage. It seems that these objectives are in conflict thus, they cannot be solved individually. Therefore, multi-objective optimisation is used to find the solution that compromises between the objectives.

II. ALGORITHM STRUCTURE

In this study, the path-planning algorithm is split into three separate modules namely Route Planner (RP), Path Planner (PP) and Coverage Planner (CP). A simple block diagram that represents each module is depicted in Fig.3 below.

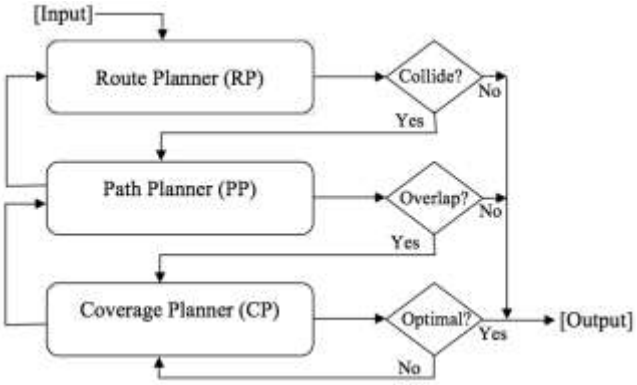

Fig 3: A structure of path-planning algorithm

Fig.3 illustrates the structure of path planning algorithm. Note that, the input to the module contains a set of essential parameters e.g., the locations of the coverage spots including start and finish points and the position of the obstacles using Global Positioning System (GPS). Based on these inputs, the algorithm generates the feasible paths at the output. Detail of the individual planner is described in Section III.

III. ROUTING PROBLEM

This section presents a method used in every planner and the integration between RP and PP. RP is a module that strategizes the complete routeing using the strategy of TSP. Ideally, the solution is routed via straight-line paths. In the event of collision, PP will seek for collision-free paths between the respective checkpoints. A series of scenarios in Fig. 4 illustrates how the original TSP route changed after avoiding obstacles.

In general, fig.4 contains four different scenarios of how the path-planning algorithm progress as the position of obstacles change. Let C_0/C_f be the start and final checkpoint and C_1 - C_7 be the set of checkpoints that the UAV must visit/monitor during the operation. The algorithm in RP uses a modified version of Genetic Algorithm (GA) proposed in [13] to fulfil the TSP rules. According to Fig.4(a), the initial solution is in the following sequence,

$$(C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7 \text{ and } C_f)$$

Without the obstacles, this sequence is considered as optimal route. However, when O_1 and O_2 are introduced as in Fig.4(b), the path between C_1 and C_2 is collided with the obstacles. With the help of PP, the optimal path between C_1 and C_2 is literally found through a tiny passage between the obstacles.

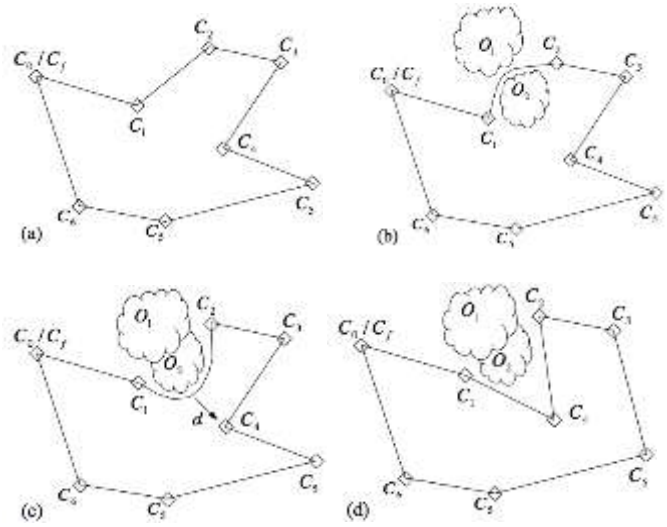

Fig.4: (a) A standard TSP route, (b) A path avoids two obstacles with a narrow passage between C_1 and C_2, (c) A path ovoids obstacle between C_1 and C_2, (d) A new route that avoid obstacles in (c).

A slightly different scenario is devised in Fig.4(c) where the obstacles O_1 and O_2 are overlapping, hence no through passage in between. Therefore, the PP would find the avoiding path either through the top or the bottom of the obstacles. Since the bottom path is shorter, it navigate through the bottom path. As the algorithm navigates along its trajectory to C_2, approximately at point *d*, the algorithm sees C_4 closer than C_2 hence, C_4 becomes a priority point over C_2. In this situation, the PP will update RP so that the original path in Fig.4(a) can be re-optimised to a new sequence,

$$(C_0, C_1, C_4, C_2, C_3, C_5, C_6, C_7 \text{ and } C_f)$$ as shown in fig.4(d).

This process is progressively optimised according to the position of the obstacles until the route is optimised and no collision.

IV. ROUTING ALGORITHM

As mentioned earlier, RP solves the TSP using GA due to the robustness-and-easy to implement. In GA's context, the checkpoints represents the genes of the chromosome. Each chromosome represents a return trip of TSP journey from starting point to the finishing point. The total genes within the chromosome depends on the number of checkpoints. A general rule is that, the *ellele* of the gene must only be swapped between genes to prevent duplicate checkpoint within the same chromosome. This property must be adhered to ensure the route visits every checkpoint once and only once.

Let's consider the nine-checkpoint example in Fig.5.

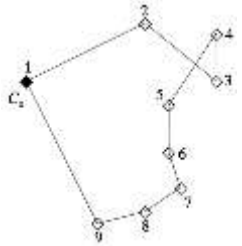

Fig. 5: An example of nine-checkpoint scenario.

Let C_0 or 1 be the fixed starting and final checkpoint, then a chromosome can be formed from the remaining checkpoints in random order, in this case the sequence is 1-2-3-4-5-6-7-8-9-1. From this chromosome, only the middle genes are subject to evolve since checkpoint 1 is fixed.

Basically, a chromosome contains a sequence of paths that connects between checkpoints (genes). As to avoid missing or duplicate checkpoints, the *ellele* must not duplicate within the same chromosome. Every evolved chromosome is evaluated using distance functions and converge estimation. In this study, the chromosome is selected randomly and evolved using single parent. From there, a new child is born using single point crossover. The mutation is done through swapping, flipping and shifting on two genes using two random insertion points within the chromosome. From this process, a new offspring that has higher fitness is created and selected as an elitist for further evaluation. This process continue until no better candidate is found.

In general, the design of the algorithm solves the generalisation of the Hamiltonian circuit in graph forms where, each execution is computed using the function in Eq. (1.0).

$$f(x) = \min \sum_{i=1}^{g} dist(x_i, x_{i+1}) + dist(x_n, x_0) \qquad (1.0)$$

Where, i=1,2,3,… n is the order in the chromosome , n is the maximum number of checkpoints.

RP is used to optimise the route using TSP, the PP determines the best avoiding path within the route if the path collides with obstacles. The PP adapts the concept of Dijkstra's algorithm [4] or known as A* (A-star). Despite the simplicity of A*, it is compatible with the grid-based spaces. Each coresponding cell in grid-based is define as vertex, V_i, where i=1,2,3,…n and n is a total available cells within the space. The cell that belong to free space is defined as $V_{free} \in V_i$ and the obstacle's cell is defined as $V_{obs} \notin V_i$. Literally, a feasible-and-collision-free path is found between V_i and the next vertex $V_i{}'$ provided $V_i{}' \in V_{free}$.

The decision about selecting the successor's state $V_i{}'$ is determined by its corresponding cost function. The function contains a set of rules that enable the transition. Let's consider a simple state transition diagram that represents a discrete path planning as shown in Fig.6.

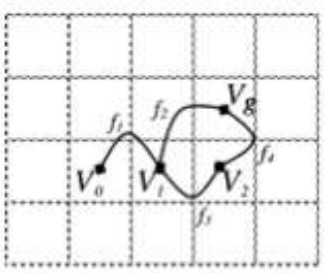

Fig.6: A simple state transition diagram.

Basically, the diagram in Fig.6 has four states/vertices labelled with $V_0, V_1, V_2, V_g \in V$ (where V_0 is an initial state and V_g is a goal state). A transition between two vertices is governed by its corresponding cost functions f_1, f_2, f_3, $f_4 \in \Sigma$. A formal definition of such transition is given by the following 5-tuple $(V, \Sigma, \delta, V_0, V_g)$, where;

- V is a finite set of states (vertices),
- Σ a finite set of function,
- δ is the transition function, that is $\delta: V \times \Sigma \rightarrow V$
- V_0 s the initial state, where, $V_0 \in V$,
- V_g is a set of goal state, $V_g \in V$.

A complete transition between V_0 and V_g has to either pass through V_1 only or alternatively through V_1 and V_2. A transition from V_0 and V_1 is taken place when f_1 is satisfied such that δ: V_0 x $f_1 \rightarrow V_1$. A transition from V_1 to V_g has two possible paths, one of which has to satisfy f_2 such that δ:V_1 x $f_2 \rightarrow V_g$, alternatively it must satisfy f_3 and f_4 such that δ:(V_1 x $f_3 \rightarrow V_2$)($f_4 \rightarrow V_g$). Supposing that, V_0 is an initial vertex, V_g is a destination vertex and V_1 and V_2 are neighbouring vertices. A transition, for example, between V_0 and V_1 is occurred when it satisfies f_1. A generic transition of the distance-based search is computed using Eq. (2.0) below,

$$f_d(dist) = \min \sum_{i=1}^{n} dist(V_{d(i-1)} V_{di}) \qquad (2.0)$$

Where i=1,2,3,…,n, n=8 represent the number of neighbouring states, whereas δ=1,2,…,m, where m is a maximum number of states before reaching V_g. The *dist* is an Euclidean distance between two vertices given by Eq. (2.1),

$$dist = \sqrt{V_{i-1}^2 + V_i^2} \qquad (2.1)$$

Adding a heuristic function to it speed ups the search as the current vertex is evaluated towards V_g. In this basis, a two-terms heuristic cost function is added into Eq.(2.0) as formalised in Eq. (2.2) and Eq. (2.3). A vertex V at neighbouring i must satisfy its corresponding function $f_\delta(dist)$.

$$f_d(dist_{heuristic}) = \min \sum_{i=1}^{g} dist_{heuristic}[(V_0, V_{di}) + (V_{di}, V_g)] \qquad (2.2)$$

Where,

$$dist_{heuristic} = G_d + H_d = \sqrt{V_0^2 + V_{di}^2} + \sqrt{V_{di}^2 + V_g^2}$$

and

$$H_d(V_{di}, V_g) \leq H_d(V_0, V_g) + G_d(V_0, V_{di}) \qquad (2.3)$$

By now, RP and PP are able to optimise the TSP route with collision free paths. In addition, CP is used to estimate the coverage if the coverage spot is covered by the obstacles (e.g., cloud, big trees). A sample work on path planning coverage estimation in agricultural field can be found in [14]. In this approach, CP is used to evaluate the closest point to the checkpoint centre or the coverage center. The selection of the point not only consider the coverage, but also the distance towards the next point as illustrated in Fig.7.

http://dx.doi.org/10.17758/EAP.EAP816005

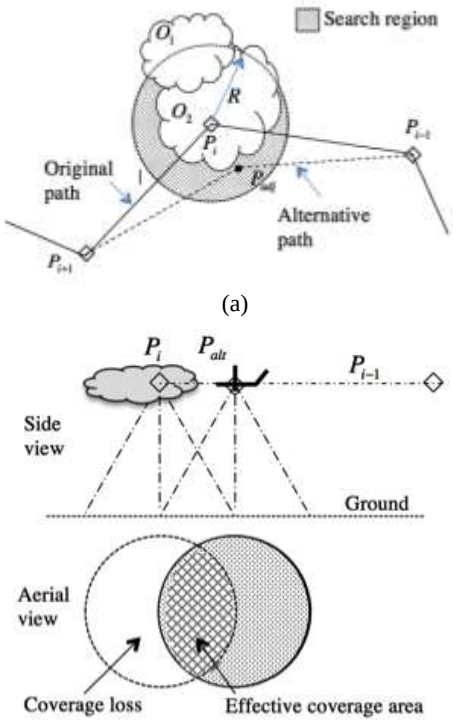

Fig.7: (a) A scenario where P_i is not accessible via original path due to obstacle O_2 (aerial view). An alternative path is found via an adjacent point P_{adj}, (b) A scenario of coverage loss due to avoiding O_2

Fig.7(a) shows that, the access to point P_i is restricted due to obstacle O_2. Therefore, CP is used to find the adjacent point P_{adj} so that it able to connect P_{i-1} and P_{i+1}. In this instance, the coverage is taken from P_{adj} instead and that causes the loss in coverage as illustrated in Fig.7(b). The farther the P_{adj} from P_i the larger the losses will be. Literally, the distance of P_{adj} to P_i determines the amount of coverage loss. As to limit the coverage loss, the position of P_{adj} must be governed by a specific radius R. Therefore, the criterion for selecting P_{adj} is important to minimise the losses. We estimate the coverage loss (CL) in term of the percentage between the obtainable coverage versus the targeted coverage. The estimation is formulated is in Eq. (3.0) below,

$$\mathrm{CL}(\%)=\frac{C_{\text{target}}-C_{\text{loss}}}{C_{\text{target}}}\mathrm{X}\ 100\% \qquad (3.0)$$

Where,

$$C_{\text{loss}}=C_{\text{target}}-C_{\text{effective}} \quad \text{and} \quad C_{\text{effective}}=C_{\text{target}} \cap C_{\text{attainable}}$$

Let C_{target} be the desired coverage area (red circle), and the $C_{\text{attainable}}$ be the attainable coverage area (blue circle). Then, the effective coverage area, $C_{\text{effective}}$ is the intersecting area (shaded) as shown in Fig.8,

From Fig.8, $\rho R^2=C_{\text{target}}$ and $\rho r^2=C_{\text{attainable}}$ respectively. Therefore, $C_{\text{effective}}$ can be expressed in Eq. (3.1),

$$C_{\text{effective}}=C_{\text{target}}(R,d_1)+C_{\text{attainable}}(r,d_2)$$

$$=r^2\cos^{-1}\left(\frac{d^2+r^2-R^2}{2dr}\right)+R^2\cos^{-1}\left(\frac{d^2+R^2-r^2}{2dR}\right)\ldots$$

$$-\frac{1}{2}\sqrt{(-d+r+R)(d+r-R)(d-r+R)(d+r+R)} \qquad (3.1)$$

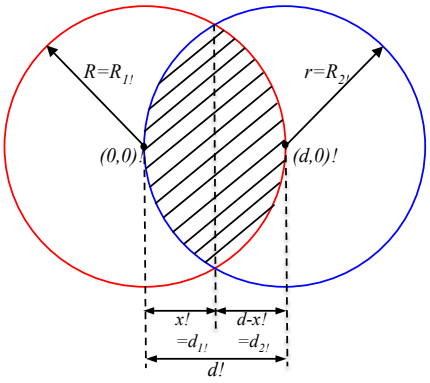

Fig.8: The circle with radius R_1 represents C_{target} and the circle with radius R_2 represents $C_{\text{attainable}}$

In the case of two identical circle, $R=R_1=r=R_2=d$, thus, $d_1=d_2=d/2$ and d represent the distance between P_{adj} and P_i. If the position of P_{adj} within the area of coverage region R_1, the Eq. (3.2) can be simplified as,

$$C_{\text{effective}}=2R^2\cos^{-1}\left(\frac{d}{2R}\right)-\frac{1}{2}d\sqrt{4R^2-d^2} \qquad (3.2)$$

Assuming R=2, then substitute R into Eq. (3.2) yields $C_{\text{effective}}$ = 4.91 and C_{desired} =12.5 respectively. Substitute these values into Eq. (3.0) yields CL%= 60.7%. That implies the effective coverage is only 39.3%. So, in this example, as long as P_{adj} resides within the radius R_1 or $d \le 1/2\ R_1$, the CL% would be less than 60.7%. That means, a smaller d (distance between P_{adj} and P_i) would enlarge the effective coverage area thereby minimise the CL. On the other hand, if P_{adj} is not found within the set value of R, the search will expand outside the boudary. This would cause the point to move away from P_i , thereby CL% would increase. Fig.9 illustrates how the CP finds the P_{adj} within the specific radius.

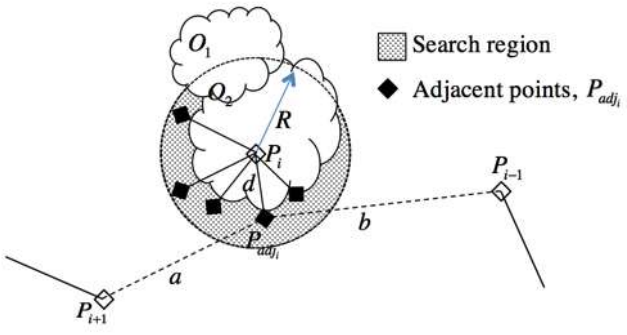

Fig.9: The illustation of a search area to locate the best adjacent point P_{adj} within the boundary R.

Since the function of CL depends very much on the distance of *d*, the position of P_{adj} that optimises the CL and the path length is given by,

$$P_{adj} = \min(a) + \min(b) + \min(d) \quad d \in R \qquad (3.3)$$

Where *a*, *b* and *d* are the relative Euclidean distances given by,

$$a = dist(P_1, P_{adj}), b = dist(P_{adj}, P_3), d = dist(P_2, P_{adj})$$

Thus,

$$P_{adj} = \min \sum_{i=1}^{m} a_i + b_i + d_i \qquad (3.4)$$

Where, $P_{adj_i} \in S$, i=1,2,3,…,m where, m is a total possible accessible point in S within R. For path length optimisation, parameter d in Eq. (3.4) has no specific condition as long as P_{adj} keeps the distance at minimal between two points. However, to improve the coverage, d has to be as small as possible.

V. RESULT

The simulation has been divided into two stages. Stage one, The algorithm consist only RP and PP to optimise the route and avoid obstacles. As such a scenario of 50 random checkpoints is created with all checkpoints scattered outside the obstacle region (red dots) defined as Scenario1. The aim of this simulation is to solve the problem described in Fig.4. The simulation has been configured to run within two-dimensional search space containing 50 x 50 cells, the escalation in height is not considered. The simulation result for Scenario1 is presented in Fig.10 below.

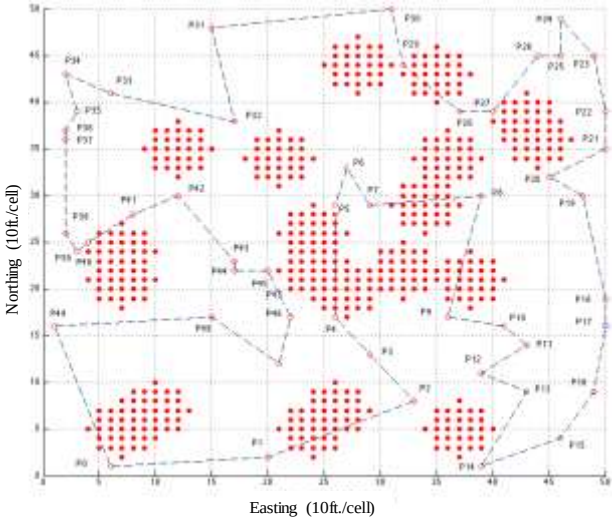

Fig.10: A standard TSP route using RP only (Scenario 1).

Fig.10 shows that, the route starts from P0 and returned to P0 as a complete return journey. At this stage, RP produces point-to-point paths with some collided paths particularly, the paths between P1 and P2 and P8 and P9. When PP get the information of the collision path, those conflicting path would be rerouted become collision-free paths as observed in Fig. 11.

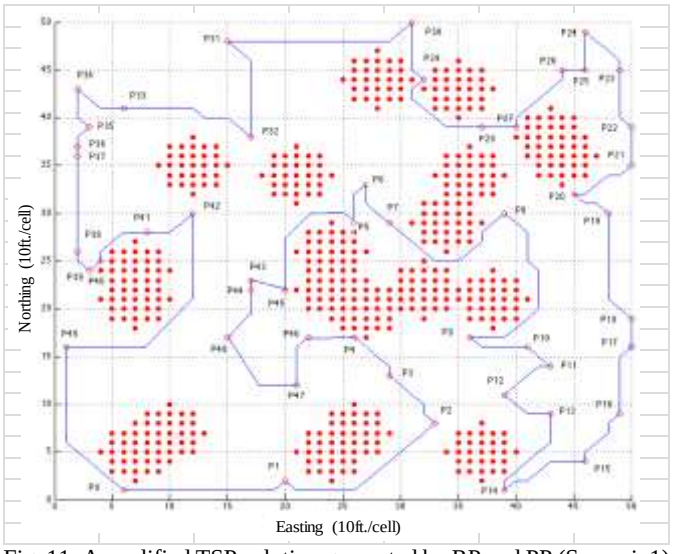

Fig. 11: A modified TSP solution generated by RP and PP (Scenario1).

It is noticed that, the solution generated by RP and PP not only free from collision, but also improve the overall route. A visual comparison between TSP and the improved route can be observed in Fig.8 below.

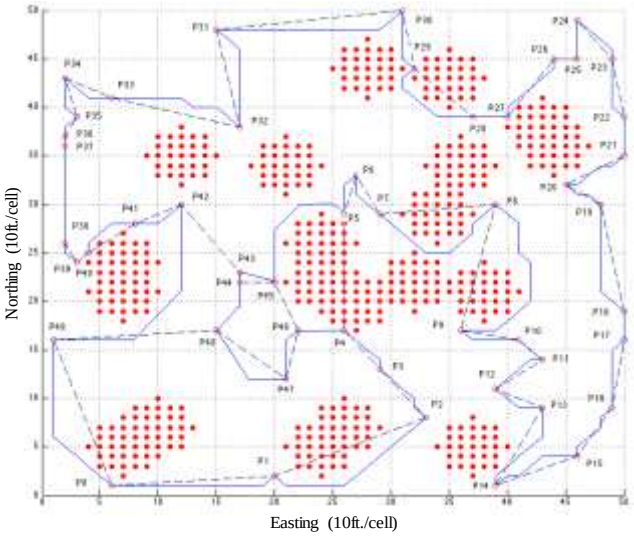

Fig.12: A comparison between the solutions obtained from RP(dotted line) and combine RP and PP(solid line) in Scenario 1.

As observed in Fig.12, the path that previously connected between P4 and P5 were rerouted to P43, P44, P45, P46, P47 and P48. This happens because, the path that avoids the obstacle between P4 and P5 found that along its way, P46 appears nearer than P5. Thus, the complete sequence is now changed to P4 through P46-P47-P48-P44-P43-P45 and finally reached P5. Essentially, this solution resembles the problem in Fig.4(c) and fig.4(d).

To summarize, the results here showed that, the interaction between RP and PP will optimise the routing provided the checkpoints are located within the free space. If the checkpoints reside within the obstacle vicinity, RP and PP may not be able to access it. Hence, CP is needed to make sure the placement of an alternative point is at the optimised location.

http://dx.doi.org/10.17758/EAP.EAP816005

To illustrate the problem, another scenario is created with 50 random checkpoints, but this time, some checkpoints were located within the obstacles regions define as Scenario2. The result of Scenario2 using RP,PP and CP is shown in Fig.9. It is observed that, P5, P11, P23, P24, P25, P33, P40, P41, P47 and P48 are located within the obstacle's region.

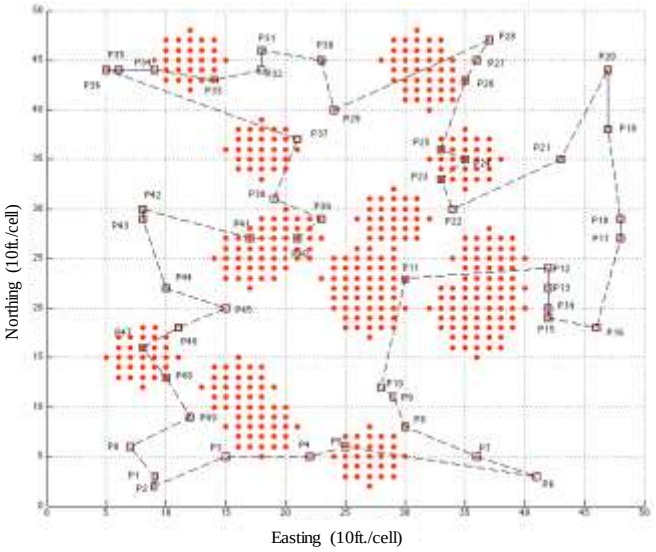

Fig.13: A routeing solution using RP only (Scenario2).

Fig.13 shows a routeing solution using RP with some checkpoints located within the obstacle's vicinity. Due to obstacles, these checkpoints are not accessible by PP. Therefore, CP is needed to search for and alternative point so that new alternative checkpoint can be passed to PP for rerouteing. As a result, the solution is then rerouted via P5p, P23p, P24p and P25p as shown in Fig.14. Similar case are observed at P33, P40, P41, P47 and P48 that are replaced by P33p, P40p, P41p, P47p and P48p respectively. This method was extended from the work in [15].

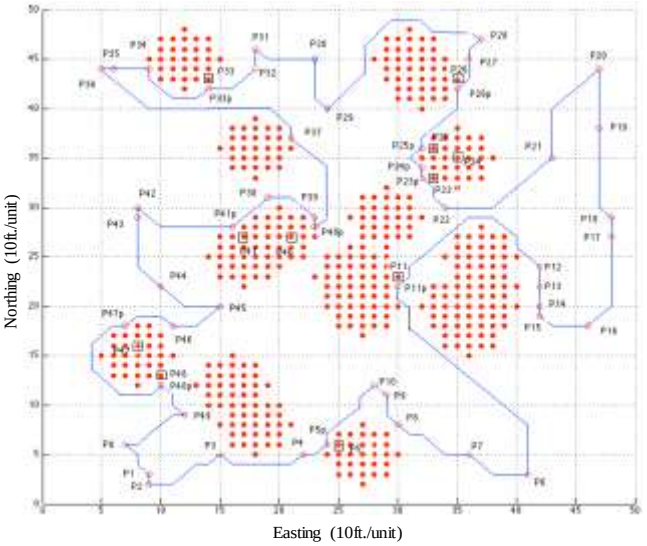

Fig.14: A new route is generated using RP, PP and CP (Scenario 2).

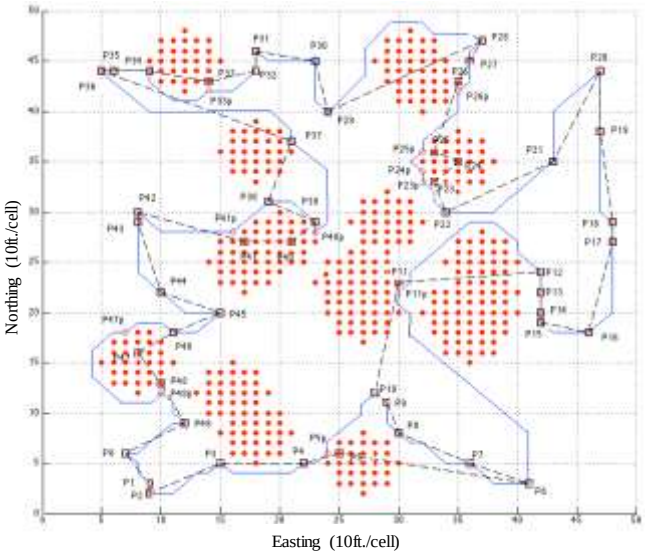

Fig.15: A comparison between the solutions obtained from RP (dotted line) and the combine algorithm (solid line) in Scenario 2.

As a matter of comparison, Fig.15 shows the result from RP versus the combined algorithm. Note that, the combined algorithm consists of RP, PP and CP that optimised the route distance as well as coverage. RP optimises the route based upon the optimal distance between previous point and the current point. It is observed that, between P5 and P11 where the path was originally connected along P6, P7, P8, P10 and P11 eventually rerouted along P10, P9, P8, P7, P6 and P11. Therefore, this solution not only avoid the obstacle, but also optimises the coverage progressively at once.

VI. MULTI-OBJECTIVE RESULT

After the optimisation process, a decision about which path is the optimal become a critical subject as the objectives are conflicting. Usually, this problem does not produce a single solution. Therefore, a set of compromised solutions have been generated by gradually increasing the value of radius *R*. Each value of R would produce different set of solution. It is observed that, if the value of *R* is relaxed, the value of CL% will increase (poor coverage). On the other hand the path length will be decreased. Since both objectives are compromised as *R* varied from 0 to 60ft. The score for both path length and CL can be visualised in Fig.16.

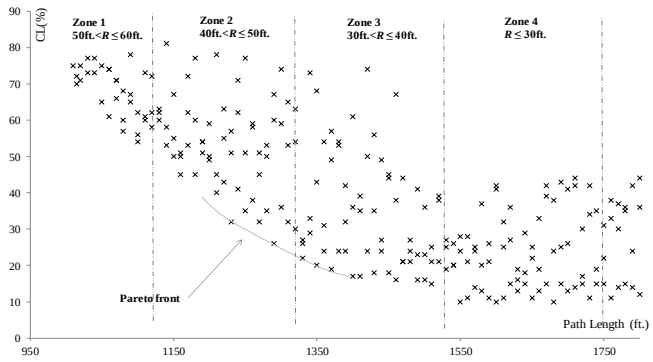

Fig.16: A trade-off between coverage loss and distance optimisation in Scenario2.

http://dx.doi.org/10.17758/EAP.EAP816005

Fig.16 shows hundreds of solutions for different value of R for each conflicting checkpoint. It is observed that, most of the solutions are dense towards the Pareto line due to minimisation effect of both objective due to *R*. This chart will provide the useful information about the outcome of each solution. If for example, the operator prefers to choose low CL% irrespective of the path length, the solution can be found within Zone 4. Whereas, if shorter path length is preferred, the solutions can be found within Zone 1. A simple weighted-sum method [16] will provide the guideline to the operator the potential solution if the weightage of each objective is known. In general, the Pareto front within Zone 2 and Zone 3 are also known as non-dominated solutions and these solution is considered as the optimal trade-off solution.

VII. Conclusion

From the result, the combination of three planners generated a set of routes that compromises between two objectives, and based on that, the operator has the right to choose the solution that fulfills their need. Thus, it would be beneficial for the operators or farmers to choose the most profitable route for their farming survey activities prior to actual operation. Future work will be focusing on many objective's optimisation that will include other objectives such as mission time, fuel consumption, risk analysis and so on.

References

[1] R. Valerdi, "Cost Metrics for Unmanned Aerial Vehicles," in in *Infotech@Aerospace*, American Institute of Aeronautics and Astronautics, 2005.

[2] P. R. Newswire, "Teal Group Predicts Worldwide UAV Market Will Total $91 Billion in Its 2014 UAV Market Profile and Forecast," 2014.

[3] International Civil Aviation Organisation, "Circular 328, Unmanned Aircraft Systems (UAS)," 2011.

[4] S. Skiena, "Dijkstra's Algorithm," *Implement. Discret. Math. Comb. Graph Theory with Math. Reading, MA Addison-Wesley*, pp. 225–227, 1990.

[5] P. Bhattacharya and M. L. Gavrilova, *Voronoi diagram in optimal path planning*. Los Alamitos: Ieee Computer Soc, 2007, pp. 38–47.

[6] L. I. Jian, "An Improved Dynamic Programming Algorithm for Bitonic TSP," 2013.

[7] S. M. LaValle and J. J. Kuffner, *Rapidly-exploring Random Trees: Progress and prospects*. Wellesley: A K Peters, Ltd, 2001, pp. 293–308.

[8] E. Masehian and D. Sedighizadeh, "A multi-objective PSO-based algorithm for robot path planning," in *Industrial Technology (ICIT), 2010 IEEE International Conference on*, 2010, pp. 465–470. http://dx.doi.org/10.1109/icit.2010.5472755

[9] W. Yanyang, W. E. I. Tietao, and Q. U. Xiangju, "Study of multi-objective fuzzy optimization for path planning," *Chinese J. Aeronaut.*, vol. 25, no. 1, pp. 51–56, 2012. http://dx.doi.org/10.1016/S1000-9361(11)60361-0

[10] F. Ahmed and K. Deb, "Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms," *Soft Comput.*, vol. 17, no. 7, pp. 1283–1299, 2013. http://dx.doi.org/10.1007/s00500-012-0964-8

[11] H. Choset, "Coverage of Known Spaces: The Boustrophedon Cellular Decomposition," *Auton. Robots*, vol. 9, no. 3, pp. 247–253, 2000. http://dx.doi.org/10.1023/A:1008958800904

[12] G. Reinelt, " The Traveling Salesman: Computational Solutions for TSP Applications," *Springer-Verlag,*, 1994.

[13] D. Goldberg and J. H. Holland, "Genetic Algorithms and Machine Learning," *Mach. Learn.*, vol. 3, no. 2–3, pp. 95–99, 1988. http://dx.doi.org/10.1023/A:1022602019183

[14] T. Oksanen and A. Visala, "Coverage path planning algorithms for agricultural field machines," *J. F. Robot.*, vol. 26, no. 8, pp. 651–668, 2009. http://dx.doi.org/10.1002/rob.20300

[15] C. Archetti, L. Bertazzi, and M. G. Speranza, "Reoptimizing the traveling salesman problem," *Networks*, vol. 42, no. 3, pp. 154–159, 2003. http://dx.doi.org/10.1002/net.10091

[16] J. Ryu, S. Kim, and H. Wan, "Pareto front approximation with adaptive weighted sum method in multiobjective simulation optimization," in *Winter Simulation Conference*, 2009, pp. 623–633 http://dx.doi.org/10.1109/wsc.2009.5429562