

Minimizing flight time and fuel consumption for airborne crop spraying

Syed Shariq Kamal¹, Patrick Jackman^{2*}, Bruce Grieve¹

(1. *University of Manchester Electronic and Electrical Engineering, Sackville Street, Manchester, M13 9PL, United Kingdom*

2. *Patrick Jackman now at Biosystems Engineering, University College Dublin, Belfield, Dublin 4, Ireland*)

Abstract: With the world's growing population and increase in urbanization, the requirement for optimized agriculture has increased. Agricultural operations such as crop spraying and water management require rigorous monitoring of crops in order to identify the correct time to spray and irrigate the crops. Thus managing vast properties require an affordable spraying strategy. Advancement in computer processing speed and algorithms has made it possible to devise such strategies to optimize several agricultural operations. One of those operations is to spray crops with pesticides and monitor crops. This requires an airborne vehicle which can monitor and spray crops efficiently. Several optimization techniques have been used in recent years to optimize the path of the aerial vehicle due to limited fuel carrying capability of these vehicles as well as the increasing cost of fuel. A comparative study has been made in this paper to analyze the performance of some of the leading techniques used to optimize agricultural operations in recent years.

Keywords: crop spraying, optimization, airborne vehicles, pesticides, artificial intelligence

Citation: S.S. Kamal, P. Jackman & B. Grieve. 2014. Minimizing flight time and fuel consumption for airborne crop spraying. *Agric Eng Int: CIGR Journal*, 16(2): 89–96.

1 Introduction

The rapidly increasing global population and diminishing resources (Pimentel, 2006) demand dynamic and efficient methods to meet agricultural productivity requirements. Crop productivity is adversely affected by several diseases which if not attended on time could cause substantial crop yield losses. Hence Jacobi (2006) has proposed a solution for the identification of fungal infection and nitrogen deficiency in wheat crop which requires daily satellite images to sense any such infections or deficiency. Similarly Kolhe et al. (2009) provides a knowledge management system for detecting crop diseases by building a database of 25 prevalent diseases of three major oil seed crops of India which are soybean, groundnut and rapeseed mustard.

The aim is to solve the disease identification and control problem. Such and many other diseases require early detection and urgent preventive measures need to be taken to save crops from infections. It has been shown by Thomson and Smith (2008) how GPS could be used for crop dusting using an agricultural airplane. Its application includes pesticide and nutrient monitoring, cotton defoliation and growth regulation. After the success of unmanned aerial vehicles in military, its application has increased in civil markets in recent years.

These vehicles can be used on-line or off-line. An on-line operation means that the vehicle could find a new way in case of meeting an unexpected obstacle in its path. An off-line operation programs the optimal path before flight and the vehicle does not have the authority to find new path while on its route (Kostas et al., 2003). The navigation of Unmanned Air Vehicles (UAV) requires path planning and optimization since the small size and limited fuel in aerial vehicles require path optimization in order to cover maximum ground with minimum amount

Received date: 2012-12-10 **Accepted date:** 2014-05-03

* **Corresponding author:** Patrick Jackman, Biosystems Engineering, University College Dublin, Belfield, Dublin 4, Ireland. Tel: 00353 17167440. Email: Patrick.jackman1@ucd.ie.

of fuel. Furthermore, it is also important to prevent these vehicles colliding with an obstacle in its course. This is a problem where the required destinations are known and there is limited time and resources to reach them and as such the route must be optimized. This is similar to The Travelling Salesman Problem which according to Applegate et al. (2006) is one of the most intensely researched problems in computational mathematics and has attracted a number of researchers from the field of computational algorithms over the years.

It is defined as a problem where a travelling salesman starting from his town has to visit each of the other towns precisely or at least once and return to his town by the shortest route (Xu, 2003). One obvious solution to the problem is to find all possible combinations in which the cities can be ordered and find the shortest possible path. However, since the computational time needed here is directly related to $(n-1)!$ possibilities where n is equal to the number of towns, the problem becomes unfeasible as the number of towns increases beyond a small number. This forced researchers to develop optimization techniques that could ensure optimum solution in minimum time. Several problem solving techniques have emerged over the past few decades but according to Applegate et al. (2006), the travelling salesman problem algorithm developed by Held and Karp in 1962 is the best known guarantee of a solution to the travelling salesman problem. It adopts a deterministic approach to minimise the number of possible combinations which consequently reduces time and guarantees the optimum solution. However, the time to find the solution increases rapidly with increase in the number of towns. Furthermore, it keeps track of the partial solutions to the problem which increases exponentially with the increase in the number of possibilities and thus requires an unfeasible amount of memory after the problem grows beyond a small number of towns.

2 Optimisation techniques

Several optimization techniques have emerged in the last few decades with an increase in processing speed which has enabled us to solve large and complex problems which were unfeasible to solve in the past.

However, dynamic programming is the only technique which guarantees an optimal solution (Applegate et al., 2006). According to Jones (2008), however Genetic algorithm is the most famous and flexible algorithm. Ferentinos et al. (2002) tested and compared Genetic algorithm and simulated annealing in application of motion planning for autonomous agricultural vehicles and concluded that both performed well and are well suited for motion planning for agricultural vehicles. Similarly Bakhtiari et al. (2011) used Ant colony algorithm for vehicle path planning in agricultural application and found excellent results. Oksanen and Visala (2009) used Greedy algorithm for path planning of field machines. It is one of the techniques which have started to make its way into agricultural applications and work is being done to improve it. Since all the above mentioned techniques are well-suited and currently being applied for agricultural applications, they have been used to solve the Travelling salesman problem in this paper to analyze and compare their performances.

2.1 Dynamic programming

Dynamic programming solves a bigger problem by breaking it down into small sub-problems. A Similar strategy is used to solve the travelling salesman problem, as mentioned in Applegate et al. (2006), Dreyfuss and Law (1977) and Conway et al. (2003) in great detail. According to Applegate et al. (2006), Dynamic programming is based on Bellman's theory which permits breaking down a larger problem into smaller sub-problems, where in an optimal tour after crossing through a number of cities the path through the remaining subset must itself be optimal. Using this idea Held and Karp (1962) presented an algorithm and solved three problems namely the scheduling problem, Travelling salesman problem and assembly-line balancing problem. Our interest however is to solve a travelling salesman problem. The aim is to greatly decrease the number of searches below the number of possible permutations. The emphasis is thus on reducing the time needed to solve the problem. Conway et al. (2003) has also proved that as the number of cities increase, Dynamic programming becomes more and more useful compared to $(n-1)!$ Combinations as it implements a Bellman

equation (Equation (1)):

$$V(x) = \max_{a \in \gamma} (x) \{F(x, a) + \beta V(T(x, a))\} \quad (1)$$

where, V is the value function; x is the current state; a is the current action from a set of available actions γ ; F is the action payoff; β is the discount factor and T is the new post action state.

It has given the basic principle of optimality of Dynamic programming represented by Equation (1) which is the basis of the results discussed later in the paper. Kennedy (1986) describes the applications of dynamic programming in farming, forestry and fisheries etc. It gives a detailed description on how it can be used for crop management. Similarly Throsby (1964) has provided a dynamic programming model for farm management and used it for decision making in specific areas of farm management. Likewise Toft and O'Hanlon (1979) has used dynamic programming for decision making on farm in drought condition in order to optimize resources. Even though a lot of work has been done in recent decades on the application of dynamic programming in agriculture, according to Throsby (1964) the number of its applications to practical situations remains relatively small. One of the reasons as mentioned by Blanco-Fonseca et al. (2011) is the limitation to the number of states/cities in a problem. Despite the proven capability of dynamic programming to reduce time by reducing the number of searches, it still takes a lot of time to solve the problem beyond a small number of states/cities.

2.2 Ant Colonies

Ant colony optimization algorithm is inspired by an organized movement of ants in group. Jones (2008) and Dorigo and Stutzle (2004) have discussed the movement of ants and the transformation of this knowledge into algorithm. Furthermore Dorigo and Gambardella (1997) have stated that artificial ant colony is capable of finding solutions to both symmetric and asymmetric travelling salesman problem by adapting to changes in environment and avoiding obstacles. According to Bell and McMullen (2004) Ant colony algorithms are quite efficient in finding good results consistently for smaller problems; however they fail to provide such good results for larger problem. Results in Bell and McMullen (2004)

show that modification improved results for larger problems as well. Bakhtiari et al. (2011) has used ant colony algorithm for optimal route planning in order to optimize agricultural operations. It has further emphasized that farming operation efficiency affects overall operational cost in agricultural system. Similarly Wang (2009) has applied a modified version of the ant colony algorithm for finding an optimal route for the distribution logistics on farms. Ant colony is being used successfully in agricultural fields to find optimal routes for land as well as aerial vehicles and modifications are being made as required. Jones (2008) has transformed the concept of movement of Ant into an algorithm for solving travelling salesman problem with the help of Equation (2) and Equation (3).

$$P = \frac{\tau(\gamma, u)^\alpha \times \eta(\gamma, u)^\beta}{\sum_k \tau(\gamma, u)^\alpha \times \eta(\gamma, u)^\beta} \quad (2)$$

$$\Delta \tau_{ij}^k = \frac{Q}{L^k} \quad (3)$$

For Equation (2): P is the probability of the k th ant making a particular move; τ is the trail level associated with that move; η is the attractiveness of a particular move.

For Equation (3): τ is the trail left by the k th ant in moving from i to j ; Q is a constant; L is the cost of that move.

Ants start off randomly in search for food and while they move they leave an invisible trail of a liquid called pheromone. These trails lead the following ants to and from the food. Here the level of this liquid is represented by τ in Equation (1), while η is the inverse of the distance of each edge in the tour and decides the whether the next step of the ant should be based more on inverse of the distance between the present location and next location or on the level of pheromone on the path. Initially, the level of pheromone is considered to be equal on all edges. Once all ants complete their tour and pheromone level is updated, next iteration is started, with the same pheromone level on the path as was at the end of the previous iteration.

2.3 Greedy algorithm

A Greedy algorithm solves travelling salesman problem locally by choosing the nearest city as the next

city at each stage. Application of Greedy algorithm for travelling salesman problem is discussed in detail by Panneerselvam (2007) and Aldous and Wilson (2000). Research is being done to use greedy algorithms to optimize several agricultural operations. Wang and Li (2012) used the Greedy algorithm for seedling transplantation where it is used to optimize the movement of a robot used for transplantation in order to minimise time and resources.

Similarly Oksanen and Visala (2009) planned the path for field machines and optimized it using the Greedy algorithm. Their results showed that there are advantages as well as disadvantages of using the Greedy algorithm for path planning; it can prove to perform well for non-omni directional vehicles. There are several ways of implementing the Greedy algorithm, Panneerselvam (2007) in particular has given 10 simple steps to program the Greedy algorithm to solve travelling salesman problem.

Even though the strategy of Greedy algorithm tends to find a local solution to the problem and does not guarantee an optimal solution, there is always a chance that the local solution found is the global solution. Intuitively, this chance would decrease as the number of cities would increase. Furthermore, the Greedy algorithm is one of the easiest algorithms to understand and implement. Jensen et al. (2004) and Ausiello et al. (1999) have given an interesting study on when the Greedy algorithm fails and whether it is efficient enough to solve the problem. Jensen et al. (2004) has given theorems that characterize the cases when the Greedy algorithm may produce a unique worst possible result. Furthermore Ausiello et al. (1999) shows on the basis of numerous results that even though Greedy algorithms rarely generate an optimal solution it is capable enough of generating good approximations of optimal solutions. However Gutin et al. (2002) has shown that the Greedy algorithm produces acceptable results for Euclidean travelling salesman problem but produces extremely poor results for symmetric and asymmetric travelling salesman problem. Also Gutin et al. (2002) has proven that the number of the cities present in the tour decide how much worse the results of Greedy algorithms would be.

2.4 Genetic algorithm

Genetic algorithms are biologically inspired algorithms similar to Ant colony optimization algorithms. According to Jones (2008) it is the most famous and flexible evolutionary algorithm which is also known as population based technique because instead operating on a single potential solution, it uses population of potential solutions. Even though the optimal result is not guaranteed, it manages to provide near approximations to optimal solutions. Similarly Noboru and Terao (1997) have used it for path planning of robotic vehicles. Both objectives were achieved and the paths were planned successfully using a Genetic algorithm. It has many more applications in agriculture than path planning. Genetic algorithms are being implemented in agricultural fields and researched for possible improvement in performance.

The three basic operators that drive the Genetic algorithm as described by Vose (1999), Man et al. (1999) and Mitchell (1996) are selection, crossover and mutation. A chromosome is a candidate solution which is a complete tour in travelling salesman problem. Kostaras et al. (2003) has shown different fitness functions according to the demands of the problem. The less the distance the fitter the tour sequence is and thus has more chance of moving into the next generation. Chatterjee et al. (1996) has described the theory and methodology of Genetic algorithms and their utilization to solve travelling salesman problem. Furthermore, it has discussed the results obtained for several two-dimensional travelling salesman problems. Three of these problems were symmetric Euclidean travelling salesman problems. Genetic algorithms produced excellent result within a reasonable time. Even though Genetic algorithms provided close to optimal solutions, it rarely provides exact optimal solutions. Results from Hu et al. (2004) and Moon et al. (2002) show that Genetic algorithms are very flexible for modification which can greatly enhance their performance.

2.5 Simulated annealing

Jones (2008) describes simulated annealing as an iterative improvement algorithm which simulates the process of annealing. Annealing is a metal casting

technique where a molten metal is heated and then cooled gradually. Simulated annealing generates randomness and avoids local minimum traps. Kirkpatrick et al. (1983) has discussed in detail the optimization using simulated annealing and how it mimics the process of annealing. Equation (4), mentioned in Jones (2008) plays a vital role in driving the algorithm to find the optimum solution. Here, is the difference between the new solution and old solution while T is the temperature and P is the probability of replacing the old solution with new solution. As the temperature decrease with time probability of finding a worse solution decreases. Similar strategy has been used to find the results presented in this paper.

$$P = \exp\left(\frac{\Delta E}{T}\right) \quad (4)$$

Simulated annealing has proved significant in path planning of vehicles in agricultural fields. Ferintinos et al. (2002) has shown a comparative study and tested simulated annealing and genetic algorithm for path planning of autonomous vehicles used on agricultural fields. It has been shown that simulated annealing outperforms Genetic algorithms. It has a far bigger scope than path planning as shown by Kuo et al. (2001) where it is used in water resource management. Several researchers have implemented simulated annealing to solve travelling salesman problem using some modifications in the generic simulated annealing.

3 Materials and methods

In this paper experiments were carried out on five optimization techniques which were programmed and tested using Matlab R2010a, Intel(R) Core(TM) i3 CPU M370 @ 2.40GHz processor, 6 GB RAM and 64 bit operating. Parameter settings for optimization techniques are as follows:

- Alpha=6, Beta=2 and Number of iterations=5 for Ant colony algorithm
- Initial Temperature=100, Decrease in temperature per iteration=0.99 for simulated annealing.
- Number of generations=300, Population size=100, Crossover fraction=1 for Genetic algorithm.

Results that follow are based upon above parameter settings and system specification.

4 Results and discussion

4.1 Performance analysis of dynamic programming algorithm

Table 1, illustrates the performance of five different optimization techniques mentioned in previous section. Table 1 shows that dynamic programming took almost 3 min to find an optimal solution to the 10 city problem. The optimum path generated by dynamic programming is illustrated in Figure 1. Table 1 also reveals that the optimum journey distance is equal to 4.3 since Dynamic programming uses a deterministic technique to ensure an optimal solution to the problem. This result would help us to gauge the performance of the rest of the techniques which do not guarantee an optimal solution to the problem. The fact that it ensures an optimal solution, gives it a slight edge over others since it eliminates the need to run the algorithm over and over again. However, as the number of cities/destinations grows beyond a small number the time it takes to solve the problem grows very large and becomes unfeasible especially in real time situations.

Table 1 The results of the five programming techniques used for solving the 10 city travelling salesman problem

Techniques	Trials No.	Distance travelled /m	Time elapsed /s	
Dynamic Programming	1	4.3	181	
	1	4.6	180	
	2	5.9	179	
	Genetic Algorithm	3	4.6	183
		4	4.3	180
Greedy Algorithm	5	4.3	178	
	1	15.3	0.2	
	2	14.5	0.2	
	Ant Colony Algorithm	3	4.3	0.2
		4	25.3	0.2
Simulated Annealing	5	5.4	0.2	
	1	4.3	0.3	
	2	4.3	0.2	
	Ant Colony Algorithm	3	4.3	0.2
		4	4.3	0.3
Simulated Annealing	5	4.3	0.3	
	1	4.3	0.2	
	2	4.3	0.2	
	Simulated Annealing	3	4.3	0.2
		4	4.3	0.2
Simulated Annealing	5	4.3	0.2	

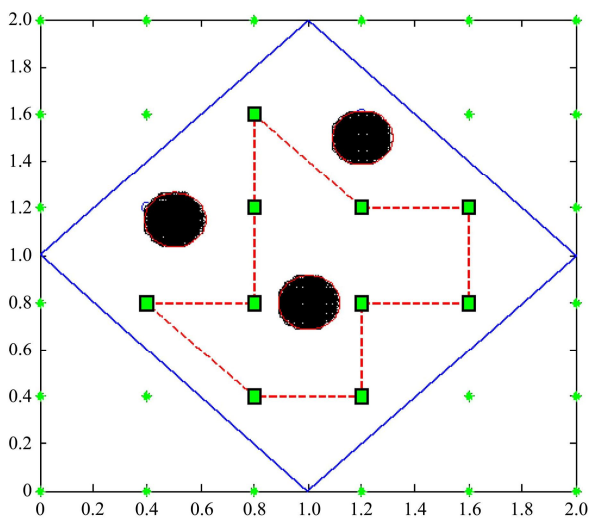


Figure 1 Illustrates travelling salesman problem solution of 10-city problem using dynamic programming

4.2 Performance analysis of ant colony optimization algorithm

Table 1 shows that it took a minimum of 0.2 and a maximum of 0.3 s to find an optimal solution to the problem. It is also important to note here that it managed to find an optimal solution five out of five times as shown in Table 1. Ant colony algorithms do not always guarantee an optimal solution however it does provide with a good solution in feasible amount of time.

4.3 Performance analysis of simulated annealing algorithm

It took simulated annealing one-fifth of a second to find the optimal solution. The results obtained for simulated annealing and the ant colony are comparable even though their working principles are different. Results show that simulated annealing works quite well for a small problem. It is famous for its working principle which efficiently finds a global optimum and ignores local optima in most cases. This is reflected in the results illustrated in Table 1.

4.4 Performance analysis of genetic algorithm

Table 1 shows that Genetic algorithm managed two out of five times in finding an optimal solution to the travelling salesman problem. Furthermore, it took the algorithm approximately 3 min to find an optimal solution. As mentioned previously, even though Genetic algorithm does not ensure an optimal solution, it is found to produce an optimal solution in most cases. The results obtained can be matched with that of the

dynamic programming with respect to the time taken. As both took approximately 3 minutes to find an optimal solution compared to other techniques, Genetic algorithm was out classed by Ant colony and simulated annealing techniques, which provided an optimal solution in all five trial and that also in fifth of a second which is 900 times less than the time taken by Genetic algorithm.

4.5 Performance analysis of greedy algorithm

Table 1 shows that the greedy algorithm managed only once to find an optimal solution to the problem. Moreover, the results obtained for other trials are far from optimum. This is because of the working principle of greedy algorithm which considers the shortest distance criterion before choosing the next destination. The problem gets worse when obstacles are introduced. There is a considerable chance of algorithm choosing a shorter distance and getting trapped in a situation where it cannot get out without intersecting an obstacle. Even though it performed the worst of all the techniques in terms of optimum path obtained it took the shortest time.

5 Conclusions

Of all the techniques discussed above, dynamic programming was the only deterministic technique used. Limitations in dynamic programming make it difficult to use it practically, even though there are ways as mentioned above in which the technique can be used successfully. Unlike heuristic techniques, it cannot give a result before it finds an optimal solution and thus consumes a lot of time. Techniques like Ant colony, simulated annealing and other heuristic techniques have the option of finding a solution after limited number of iterations, even though it does not ensure an optimal solution, in most cases it manages to find a good solution. This option can be used where good solutions are required quickly. Unlike dynamic programming heuristic techniques require tuning of parameters. All parameters have to be changed with the unique features of a problem. So for example, for a larger problem one might have to increase the number of iterations or in case of Genetic algorithm the number of generations and number population in order to find an optimal or near to optimal result. These parameters are tuned with

experience and trial and error method. This therefore requires an offline tuning of the algorithms before using it on-line in cases where parameters cannot be changed on-line.

Results discussed in section 3 and based on the discussion above make it quite clear that Ant colony and simulated annealing have the decisive advantage over others for a 10 destination problem that is classed as a small problem. However, we have to say that dynamic programming performed better than the genetic algorithm by guaranteeing an optimum solution in the same amount of time and therefore should be rated as third best,

whereas greedy algorithm performed worst with most of the solutions going into local minima. However, since this study shows and compares the result for a small problem, conclusions cannot be made for larger problems with more number of destinations. Furthermore, the time taken to find the result can vary since it largely depends on programming and system used for simulation. However, performance can be compared based on the results. These results will help farmers in choosing the right programming technique for the unmanned aerial vehicle before it is used for crop dusting and other agricultural operations.

References

- Aldous, J. M. and R. J. Wilson. 2000. Graphs and applications an introductory approach, Springer, New York, NY, USA.
- Applegate, D. L., D. Bixby, V. Chvatal, and W. Cook. 2006. The Travelling Salesman Problem: A Computational Study. Princeton University Press, Princeton, NJ, USA.
- Ausiello, G., P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. 1999. Complexity and Approximation Combinatorial Optimization Problems and Their Approximability Properties, Springer, Berlin, Germany.
- Bakhtiari, A. A., H. Navid, J. Mehri, and D.D. Bochtis. 2011. Optimal route planning of agricultural field operations using ant colony optimization. *Agricultural Engineering International: CIGR Journal*, 13(4): 1-16.
- Bell, J.E. and P.R. McMullen. 2004. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18(1): 41-48.
- Blanco-Fonseca, M., M. Himics, M. Henseler, and S. Helaine. 2011. Dynamic optimization problems: different resolution methods regarding agriculture and natural resource economics. In: *Bio-Economics Models applied to Agricultural Systems*, Springer, New York, NY, USA. pp. 29-57.
- Chatterjee, S., C. Carrera, and L. A. Lynch. 1996. Theory and methodology Genetic algorithms and travelling salesman problems. *European Journal of Operational Research*, 93(3): 490-510.
- Conway, R. W., W. L. Maxwell, and L. W. Miller. 2003. Further problems with one operation per job. In: *Theory of Scheduling*. United States of America: Courier Dover Publications, Mineola, NY, USA. pp. 61-65.
- Dorigo, M. and T. Stutzle. 2004. Ant colony optimization, MIT Press, Cambridge, MA, USA.
- Dorigo, M. And L. M. Gambardella. 1997. Ant colonies for the travelling salesman problem. *BioSystems*, 43(2): 73-81.
- Dreyfuss, S. E. and A. M. Law. 1977. The art and theory of dynamic programming. Academic Press, New York, NY, USA.
- Ferentinos, K. P., K. G. Arvanatis, and N. Sigrimis. 2002. Heuristic optimization methods for motion planning of autonomous agricultural vehicles. *Journal of Global Optimization*, 23(2): 155-170.
- Gutin, G., A. Yeo, and A. Z. Verovitch. 2002. Travelling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. *Discrete Applied Mathematics*, 117(1-3): 81-86.
- Held, M. and R. M. Karp. 1962. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(3): 196-210.
- Hu, X.-B., S. F. Wu, & J. Jiang. 2004. On-line free flight path optimization based on improved genetic algorithms. *Engineering Applications of Artificial Intelligence*, 17(8): 897-907.
- Jacobi, J. 2006. Identification of fungal infection and nitrogen deficiency in wheat crop using remote sensing. *Geoscience and Remote Sensing Symposium*, pp.664-667.
- Jensen, J. B., G. Gutin, and A. Yeo. 2004. When the greedy algorithm fails. *Discrete Optimization*, 1(2): 121-127.
- Jones, M.T. 2008. Biologically inspired and hybrid models. In: *Jones, M.T. Artificial Intelligence A systems Approach*. Jones and Bartlett Learning, Hingham, MA, USA. pp. 423-430.
- Kennedy, J. O. S. 1986. Dynamic programming applications to agriculture and natural resources, Elsevier, Amsterdam, Netherlands.
- Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by simulated annealing. *Science*, 220(4598):

- 671-680.
- Kolhe, S., H.S. Saini, R. Kamal, and G.K. Gupta. 2009. KMSCD: Knowledge Management System for Crop Diseases, World Congress on Nature and Biologically Inspired Computing, pp.812-817.
- Kostasas, A. N., I. K. Nikolos, N. C. Tsourveloudis, and K. P. Valavanis. 2003. Evolutionary algorithm based offline/online path planner for UAV navigation. *IEEE Transactions on Systems, Man and Cybernetics*, 33(6): 898- 912.
- Kuo, S. F., C. W. Liu, and G. P. Merkley. 2001. SW—Soil and water: application of the simulated annealing method to agricultural water resource management. *Journal of Agricultural Engineering*, 80(1): 109-124.
- Man, K.F., K.S. Tang, and S. Kwong. 1999. Genetic Algorithms Concepts and Designs, Springer, New York, NY, USA.
- Mitchell, M. 1996. An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, USA.
- Moon, C., J. Kim, G. Choi, and Y. Seo. 2002. An efficient genetic algorithm for the travelling salesman problem with precedence constraints. *European Journal of Operational Research*, 140(3): 606-617.
- Noboru, N. and H. Terao. 1997. Path planning of an agricultural mobile robot by neural network and genetic algorithm. *Computers and Electronics in Agriculture*, 18(2-3): 187-204.
- Oksanen, T. and A. Visala. 2009. Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*, 26(8): 651-668.
- Panneerselvam, R. 2007. Design and Analysis of Algorithms, Prentice Hall, New Delhi, India.
- Pimentel, D. 2006. Soil erosion: A food and environmental threat. *Environment, Development and Sustainability*, 8(1): 119-137.
- Thomson, S. J. and L. Smith. 2008. Crop dusting using GPS. *IEEE Aerospace and Electronic Systems*, 23(3): 14-17.
- Throsby, C.D. 1964. Some dynamic programming models for farm management research. *Journal of Agricultural Economics*, 16(1): 98-110.
- Toft, H.I. and P.W. O'Hanlon. 1979. A dynamic programming model for on-farm decision making in a drought. *Review of Marketing and Agricultural Economics*, 47(1): 5-16.
- Vose, M. 1999. The Simple Genetic Algorithm, MIT Press, Cambridge, MA, USA.
- Wang, L. 2009. An improved ACO algorithm to optimize the agricultural commodity distribution routing under B to C environment. *International Conference on Test and Measurement*, 2, pp. 225-228.
- Wang, Y. and T. Li. 2012. Optimization of seedlings transplanting strategy based on greedy algorithm. *Advances in Intelligent and Soft Computing*, 123(1): 33-39.
- Xu, J. 2003. Theory and application of graphs. Springer, New York, NY, USA.