Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria (INIA)
Available online at www.inia.es/sjar
http://dx.doi.org/10.5424/sjar/2013113-3635
Spanish Journal of Agricultural Research 2013 11(3): 603-614
ISSN: 1695-971-X
eISSN: 2171-9292

# A hybrid genetic algorithm for route optimization in the bale collecting problem

C. Gracia[1]*, B. Diezma-Iglesias[2] and P. Barreiro[2]

[1] *Universitat Politècnica de València. Camino Vera, s/n. DOE. 46022 Valencia, Spain*
[2] *Universidad Politécnica de Madrid. ETSI Agrónomos. Ciudad Universitaria. 28040 Madrid, Spain*

## Abstract

The bale collecting problem (BCP) appears after harvest operations in grain and other crops. Its solution defines the sequence of collecting bales which lie scattered over the field. Current technology on navigation-aid systems or auto-steering for agricultural vehicles and machines, is able to provide accurate data to make a reliable bale collecting planning. This paper presents a hybrid genetic algorithm (HGA) approach to address the BCP pursuing resource optimization such as minimizing non-productive time, fuel consumption, or distance travelled. The algorithmic route generation provides the basis for a navigation tool dedicated to loaders and bale wagons. The approach is experimentally tested on a set of instances similar to those found in real situations. In particular, comparative results show an average improving of a 16% from those obtained by previous heuristics.

**Aditional key words**: precision agriculture; logistics; wheat harvest.

## Introduction

An increasing pressure in sustainability and production quality requirements is demanding evident managerial changes in agriculture. Farming tasks need the introduction of more advanced monitoring and information systems to secure compliance with stipulations and standards. The development of affordable positioning systems using satellites, along with the general packet radio service, universal mobile communications and communication systems have led to a fast development of telemetry and vehicle location systems in agricultural contexts. Agricultural industry is now capable of collecting more comprehensive data that allows the application of methods in order to improve the management of agricultural tasks involving the coordination of machines and vehicles. Such technologies can provide accurate information to make reliable plans to be used as a part of the decision support system in a farm management information system (FMIS) comprising information management methods tied to the automation of human decision making as shown in Hameed *et al*. (2012).

Precision agriculture (PA) is conceptualized by a system approach to re-organize the total system of agriculture towards a low-input, high-efficiency, sustainable agriculture (Cook & Bramley, 1998). PA benefits from a suite of technologies, such as global positioning system (GPS), geographic information system, automatic control, in-field and remote sensing, miniaturized computer components, mobile computing, advanced information processing, and telecommunications (Zhang *et al*., 2002). Recently, web-based architecture approaches have been developed (Nikkilä *et al.*, 2010) to the implementation of FMIS fulfilling PA requirements such as storing sensor data, creating the operation plans and providing support in everyday farm activities (Sorensen *et al*., 2011). Some major field operations are performed throughout the planned coordination of different farm equipment comprising self propelled and/or towed machinery. In most cases, a main unit is responsible for performing the task itself,

* Corresponding author: cargraca@omp.upv.es
Received: 16-10-12. Accepted: 24-06-13.

Abreviations used: BCP (bale collecting problem); CVRP (capacited vehicle routing problem); FMIS (farm management information system); GA (genetic algorithm); GPS (global positioning system); HGA (hybrid genetic algorithm); NNH (nearest neighbour heuristic); NP (nondeterministic polynomial time); PA (precision agriculture); TSP (travelling salesman problem); VRP (vehicle routing problem).

while one or more support units accomplish service tasks (Bochtis & Sorensen, 2009; Bochtis & Sorensen, 2010). The bale collecting problem (BCP) appears after mowing and harvest operations in grain and consists of defining the sequence in which bales spread over the field have to be collected. Once mower conditioners and grain harvesters have operated throughout the field, the straw is left behind in rows in order to be compressed and compacted into either cylindrical or prismatic packages by cylindrical or prismatic balers. These are easy to handle and transport. Bales remain scattered on the surface of the field awaiting their collection by loaders and further transportation in wagons (either self-propelled or pull-type) to store them in silos, silage bunkers and barns. The BCP is about operations involving the collaborative work of several machines and vehicles. Therefore, planned management becomes necessary to coordinate the various tasks efficiently. Usually the sequencing of collection is decided by the operator himself based on his skills and experience, this leads to decisions based on simple on the go assignment rules which in the case of complex problems are way far from optimal solutions and involve a corresponding loss of efficiency. On the other hand, an accurate bale collecting plan and its proper execution are achievable. Balers, loaders and bale wagons can be provided with positioning system based devices enabling geo-referenced information (Amiama *et al*., 2008) which makes possible to know the exact allocation of bales as well as tracking a vehicle from a predetermined path. The BCP can be modelled and solved efficiently by applying optimization techniques, and thereafter be integrated as a part of a decision support system within a FMIS.

To solve the BCP, there will have to be determined the routes, which will be subsequently followed by loaders and machinery. This leads to the application of criteria taking into account issues such as minimizing non-productive time, fuel consumption, or distance travelled, which may result in significant economic benefits as well as in environmental benefits. Hence, it becomes necessary to describe such operations by mathematical models that can be used for optimal allocation, route planning and timing. Parameters regarding vehicles and machines (travelling speed, capacity, unloading and loading time, operating performance, etc.), plots (geometry, presence of obstacles, biomass production volumes, etc.), silos (position, capacity, etc.) make such modelling difficult.

The BCP belongs to a class of Operations Research problems known as vehicle routing problems (VRP) which were first introduced in Dantzig *et al.* (1954) and have been widely studied since. Eksioglu *et al*. (2009) have developed a taxonomic review for the classification of the abundant literature published on this problem. Despite the fact that field tasks involve the collaborative use of vehicles, only recently there have been transferred these concepts to the agricultural environment (Bochtis *et al*., 2013).

VRP can be modelled in terms of mathematical programming. According to the theory of computational complexity, most of them are nondeterministic polynomial time complete (NP-Complete) (Garey & Johnson, 1979). Procedures which have been proposed usually focus on the use of algorithmic methods based on the application of meta-heuristics. The use of meta-heuristics methods representing successful animal team behaviour has been extended, *i.e.,* particle swarm optimization inspired in birds flocks or fish schools (Kennedy *et al*., 2001), artificial immune systems (Dasgupta, 1999; De Castro & Timmis, 2002), optimized performance of bees (Baykasoglu *et al.*, 2007), ant colony optimization (Dorigo & Stützle, 2004) and genetic algorithms (GAs) (Holland, 1975; Goldberg, 1989).

This paper presents a hybrid genetic algorithm (HGA) to efficiently solve the BCP appearing after mowing and harvesting operations. Such techniques have been already used successfully in industrial vehicle routing problems (Baker & Ayechew, 2003). The algorithmic route generation will provide the basis for a navigation tool dedicated to loaders and bale wagons, so that it will increase the overall field efficiency of collection operations. The low computational requirements of the proposed method make feasible its implementation for large scale operations.

## Material and methods

### Mathematical model

As in the VRP, the BCP can be represented as a graph theoretic problem. Let $G = (N,A)$ be an undirected graph. The node set N corresponds to the position of the set of bales B from 1 to n in addition to the silo or barn position numbered as 0 ($N = \{0,1,\ldots,n\}$). A unitary demand $q_i = 1$ of bales has been assigned to each position node i ($1 \leq i \leq n$). Moreover, $A = \{(i,j) / i,j \in N; i < j\}$ represents the set of the $n(n+1)/2$ existing edges connecting the n + 1 nodes. Each of these edges has an associated aprioristic cost, $c_{ij} > 0$, which represents the cost of sending a vehicle from node i to node j. These $c_{ij}$ are assumed to be symmetric ($c_{ij} = c_{ji}$, $0 \le i,j \le n$), and proportional to the Euclidean distance, $d_{ij}$, between the two nodes. The collection process is to be carried out by a fleet of V vehicles ($V \ge 1$) with equal capacity, $K \ge \max\{q_i / 1 \le i \le n\}$. Notice that each vehicle could be a couple loader-transporting wagon or an auto-loader. The problem is to determine the exact tour for each vehicle so that the total travelled cost is minimised. Each vehicle is linked only to one tour because of modelling purposes. Some additional constraints associated with the problem are the following: (i) each non-silo node is supplied by a single vehicle, (ii) all vehicles begin and end their tours at the silo (node 0), (iii) a vehicle cannot stop twice at the same non-depot node, (iv) no vehicle can be loaded exceeding its maximum capacity.

The only decision variable is $X^v_{ij}$:

$$X^v_{ij} = \begin{cases} 1 \text{ if vehicle v drives from mode } i \text{ to mode } j \\ 0 \text{ otherwise} \end{cases} \quad [1]$$

The objective function of the mathematical model is:

$$\min \sum_{v \in V} \sum_{(i,j) \in A} c_{ij} X^v_{ij} \quad [2]$$

subject to

$$\sum_{v \in V} \sum_{j \in N} X^v_{ij} = 1 \quad \forall i \in B \quad [3]$$

$$\sum_{i \in B} q_i \sum_{j \in N} X^v_{ij} \le K \quad \forall v \in B \quad [4]$$

$$\sum_{J \in N} X^v_{0j} = 1 \quad \forall v \in V \quad [5]$$

$$\sum_{i \in N} X^v_{ik} - \sum_{j \in N} X^v_{jk} = 0 \quad \forall k \in B \text{ and } \forall v \in V \quad [6]$$

$$X^v_{ik} \in \{0,1\}, \ \forall (i,j) \in A \text{ and } \forall v \in V \quad [7]$$

Eq. [3] is to make sure that each bale is assigned exactly to one vehicle. One arc form bale position i is chosen, whether or not the arc goes to another bale position or to the silo. Eq. [4] states capacity constraints, so that the sum of all bales collected by a vehicle has to be less than or equal to the loading capacity of the vehicle (transporting wagon). Finally flow constraints are shown in Eqs. [5] and [6] where it is guaranteed that each vehicle will leave the silo once and that the number of vehicles entering every bale k and the silo must be equal to the number of vehicles leaving. There will be a lower bound on the number of vehicles necessary to collect all the bales on the field,

$$V_{\min} = \left\lceil \frac{\sum_{i \in B} d_i}{K} \right\rceil .$$

## Solution approach

Most VRP are NP-hard, and so is the BCP, this explains why most research efforts have focused on heuristics. Various approaches to solve the classical VRP have been investigated over the past decades. These range from the use of pure optimization methods for solving small size problems to the use of heuristics and meta-heuristics that provide near-optimal solutions for medium and large-size problems with complex constraints (Cordeau *et al.*, 2002; Toth & Vigo, 2002). Most of these methods focus on minimizing an aprioristic cost function subject to a set of well-defined constraints. However, and because real-life problems are complex enough so that not all possible constraints, costs and desirable solution properties can be considered in advance, there is a need for methods capable to provide a large set of alternative near-optimal solutions, so that decision-makers can choose among them according to their specific necessities and preferences.

In most meta-heuristics, each stage (iteration) of the search algorithm starts with a solution (or set of solutions). In the next stage a new candidate (or set of candidates) is evaluated within the local space of the previous solution. The evaluation will estimate the performance of the new candidate and compare with the performance reached in the previous stages. Based on this evaluation, the candidate or candidates can be accepted, becoming part of the solution for that stage, or rejected, in which case the solution is maintained. The process is repeated until certain stopping criteria are met.

In a GA, a population of chromosomes which encode candidate solutions (individuals) of the problem evolves toward better solutions. Goldberg (1989) summarizes the attributes of GAs. Sometimes, because of constraints in real problems, it is very difficult for a pure GA to effectively explore the solution space. In those cases it is advisable to combine it with some sort of heuristic that will guide the local search optimization. Hybrid approaches frequently have better results than either method independently, and thus numerous successful practices incline to use a hybrid approach (Chen *et al.*, 2008; Gracia *et al.*, 2013). The main obstacle when developing a GA is to create an effective encoding so that genetic operators can be applied while the obtained solutions are feasible and do not violate the problem constraints.

In such case, the proposed hybrid GA to solve the BCP will have the features described in the following paragraphs.

*Parameter settings*

The following input parameters need to be defined to run the GA: population size (Pop); maximum number of iterations (iter) and stopping criteria; size of cloning proportion (Elite) maintained through iterations; and proportion of individuals generated by mutation (Mut) and crossover (Xover) according to the following expression:

$$1 = Ellite + Mut + Xover \quad [8]$$

*Genetic encoding*

A solution for BCP will consist of a number of V ordered sequences (one for each tour to plan) each of which containing some different nodes form set N. Several encodings have been used for VRP in literature. Sometimes solutions are represented as binary strings, but that kind of representation does not suit well to BCP.

It is not difficult to specify the number of vehicles and which bales are inside each vehicle but it becomes too intricate when the order of the bales needs to be given. Using the order of bales instead of binary values solves the problem. Therefore an efficient encoding to represent the solutions, which is easily applicable to the BCP, is to define a solution as a pair of vectors. The first vector (sequence vector) contains a permutation of n elements that represents the ordered sequence that will reflect all the different bales (B) to be collected. The second vector (breakpoints vector) contains the position of V-1 elements from the above sequence vector delimiting the different tours.

As an example, to illustrate the genetic encoding, suppose a BCP where the tours begin and end at a silo location identified as node 0, nine bales have to be collected (n = 9) and they are scattered on a plot in different locations identified by different integer numbers (1 through 9). Suppose the maximum load capacity of each transporting wagon three bales (K = 3), which means that the minimum number of trips to do is three (Vmin = 3). One arbitrary solution will be defined by its sequence (S) and breakpoints (R): S = [4 6 5 9 2 3 7 8 1], R = [3 7], this will to the following paths: path1 = [0 4 6 5 0]; path2 = [0 9 2 3 0] path3 = [0 7 8 1 0]. Notice that because all demands are equal to 1, for BCP breakpoint vector will be the same for all solutions.

*Initial population generation method*

The way individuals belonging to initial population are generated is of great importance to the performance of the algorithm, since it contains most of the elements the final best solution is made of. Sometimes individuals are randomly generated, but initial population may be also obtained from other constructive methods. It is called seeding when solutions from other algorithmic techniques join the randomly chose solutions in the population. A requirement for the good performance of the proposed approach is the generation of a wide variety of initial solutions. Similar to that proposed for VRP in Wang & Lu (2010), to create a well-structured and diverse initial population to solve the BCP, the production of initial individuals proposed results from a random creation method in combination with the nearest neighbour constructive heuristic (Jünger *et al*., 1995) and the incorporation of the nearest addition method (Bentley, 1992) into the sweep algorithm (Gillet & Miller, 1974). Therefore the initial population is expected to globally explore the space of solutions so that capability of the GA is improved.

The random generation of solutions consists of random permutation sequence of n nodes (all except the silo node). Further breakpoint vectors are obtained by partitioning chromosomes into segments under the vehicle's capacity constraint.

The nearest neighbour constructive heuristic was first used to determine a solution to the travelling salesman problem (TSP). Sequence solution is constructed by adding the nearest node from current position of the vehicle. This algorithm suits quite well the common assignment rule followed by an experienced operator when collecting the bales spread over a field. The solution obtained by this heuristic will be used further as a lower bound to estimate the efficiency of the HGA approach. The steps of the algorithm can be summarized as follows:

— Step 1. Set count = 0.
— Step 2. Stand on the silo vertex as current vertex.

Set available capacity $c = K$ (maximum capacity of the vehicle).

— Step 3. Find out the shortest edge connecting current vertex and an unvisited node i.

— Step 4. Set current vertex to i. Mark i as visited. Set $c = c - 1$ and count = count + 1.

— Step 5. If all the vertices in domain are visited, then terminate.

— Step 6. If $c = 0$ then set breakpoint = count and go to step 2. If $c > 0$ go to step 2.

The ordered sequence of the nodes (excluding the silo node) together with the sequence of breakpoints are the two output vectors encoding solution provided by the algorithm.

The sweep method, or Gillet and Miller algorithm, belongs to a class of heuristic algorithms called Cluster first, Route second. In this type of algorithm nodes are grouped in clusters. Later it is optimized the way nodes belonging to each cluster are sequenced. The algorithm starts from the polar coordinates ($r_i$, $\theta_i$) of all nodes relative to the silo node which is adopted as the origin of these ($r_i = 0$). The construction of one tour begins with the union of the origin node to an arbitrary node and the remaining nodes of the chromosome are determined in terms of angle increases of sweep. The tour covers as many points as the vehicle's capacity constraint allows. The process is completed when all the points of the system have been swept. Each group of nodes forming a whole tour will later use another algorithm to generate the sequence the vehicle has to follow.

As described in Wang & Lu (2010), the following steps depict the nearest addition method into the sweep algorithm:

— Step 1. Calculate the coordinates of all nodes relative to the silo (X,Y)

$$\begin{cases} X_i = x_i - x_0 \\ Y_i = y_i - y_0 \end{cases} \quad [9]$$

where ($X_i$,$Y_i$) are the coordinates of the ith node relative to the silo node ($x_0$,$y_0$), and ($x_i$,$y_i$) are original coordinates of the ith node.

— Step 2. Calculate polar coordinates ($r_i$,$\theta_i$) of nodes from relatives coordinates obtained in Step 1.

— Step 3. Sort the nodes in ascending order of their polar angles.

— Step 4. Generate the structured population. The nodes permutations are determined on the sorted $\theta_i$. Given n nodes, a total of n individuals, each starting at a different node, are generated. Every of the n sequences are partitioned into different segments according to vehicle's capacity constraint.

— Step 5. Strengthen the chromosome structures. The tours are improved using the nearest addition method described in Bentley (1992). Within each tour, the sequence of nodes is constructed by the nearest neighbour heuristic starting at the silo node.

### *Fitness value*

In order to perform a natural selection, each individual is evaluated in terms of its fitness value which is obtained by a fitness function. The fitness value weighs the quality of a solution and enables to compare it to other ones. The total distance travelled will be the fitness value used. The shorter the distance, the more efficient the solution is.

### *Crossover procedure*

The main genetic operator is crossover, which simulates a reproduction between two parent solutions. It recombines them in a certain way generating one or more children solution. The children share some of the characteristics of the parents which are passed through future generations. However it is not able to produce new characteristics.

A common recombination operator is the Simple Crossover which chooses a random cut to divide each parent in two strings. Children are generated by exchanging parents' strings. In BCP the only difference between an individual and another is the order in which permutes the elements of the chromosome. It makes no sense to talk of pure recombination since the association of different parents would generate infeasible solutions (double some positions and lack in others). In this case, it suites better an operator such as the linear order crossover operator (Davis, 1985). In linear order crossover two breaking points are selected randomly, the elements between these points are copied to the same positions of the offspring. The copied elements are deleted from the other parent, and the remaining symbols are inherited, beginning with the first position following the second crossover point.

Selection of individuals for crossover is made through a fitness proportionate selection, which is a frequently used method in which each individual is assigned an occurrence probability proportional to its fitness value.

*Cloning and mutation procedures*

Mutation is applied to a single solution with a certain probability. It makes small random changes in the solution adding some new characteristics gradually. In order to have greater dispersion in the new individuals generated, three different mutation operations are proposed here: swap procedure, sliding procedure and 2-Opt movement procedure, shown in Fig. 1. Such mutation procedures have been widely applied when solving the TSP (Brady, 1985; Martin *et al*., 1991). To apply all three mutation operations, there will first have to be selected two random positions from the sequence vector of the individual. The swap procedure consists of exchanging the elements of those two positions. The sliding procedure glides all elements contained between selected positions one position towards the left. The 2-Opt movement belongs to local search algorithms. According to Gendreau *et al*. (1998) combining local search algorithms with GA is necessary to solve VRP efficiently. Most local search heuristics can be described as Lin's λ-Opt algorithm (Laporte *et al*., 2000). The algorithm removes λ edges from the tour and the remaining segments are connected in every other possible way. The 2-Opt algorithm removes two edges from a tour and reconnects the resulting subtours in the other possible way as shown in Fig. 1.

Elitism maintains certain individuals from one generation to the next one by cloning them. Cloning and mutation processes are made through the following steps: (1) random positions are assigned to all initial elements of the population; (2) all individuals in population are grouped in equal sets; (3) in each set it is chosen the individual with the highest fitness value and it is cloned to take part in the next generation population (elitism); (4) the three mutation procedures are applied to the individual chosen in phase 3; (5) new three individuals take part in the next iteration population. Notice that according to this procedure, mutation rate (Mut) will be three times the elite rate (Elite) so for the HGA proposed here Eq. [8] can be rewritten as:

$$1 = 4 * Ellite + Xover \qquad [10]$$

Fig. 1 shows an example of mutation procedures proposed offer from a sequence [1 2 3 4 5 6 7], two positions randomly selected {2} and {6}.

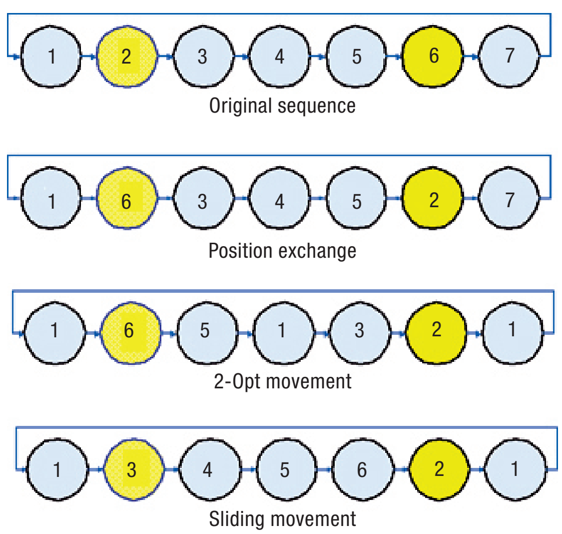


**Figure 1.** Description of three different basic mutation operations.

## Experimental study

The proposed algorithm has been implemented in commercial software MATLAB® release R2007b. The set of input parameters for the GA were {Pop = 80, Iter = 6,000, Elite = 0.125, Xover = 0.5, Mut = 0.375} these setting rates are similar to those determined in Wang & Lu (2010), where it is employed the response surface methodology to conduct systematic experiments with various crossover and mutation probabilities, so that the optimal combination of these parameters when solving the CVRP are determined.

In order to test the proposed HGA approach, a computational experiment based on realistic scenarios has been conducted. Each scenario is generated by a novel BCP generator from a certain test instance. Results obtained are later compared to other heuristic approaches. In Grisso *et al*. (2007) it is addressed a straightforward instance problem; the performance of the HGA for that instance is also discussed.

## Instances generation

Because there are not benchmark problems for the BCP, and in order to test the proposed algorithm under several realistic situations, it is required to develop a problem generator able to generate problem instances from a certain set of parameters. A problem will be defined by the capacity constraint of the vehicle (K) and by the n + 1 exact locations in a field: n corresponding to the collecting bales, and one corresponding to the starting/ending point of the tour: a silo, the

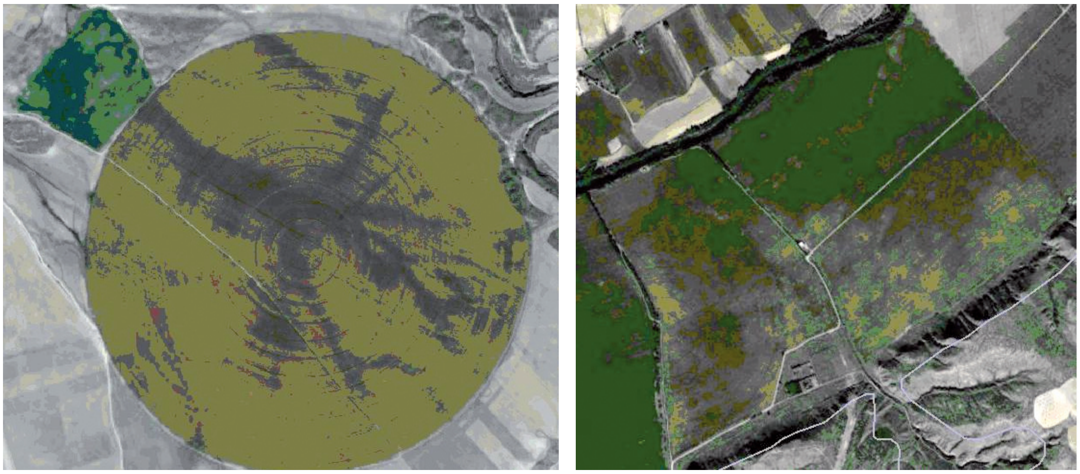

**Figure 2.** Examples of yield variation in field for different plot shapes in Toledo Province (Spain). *Source:* SIGPAC (www.magrama.gob.es).

vehicle's entry point to the field, etc. Therefore the problem generator will have to be able to generate the position at which bales are located.

If we consider a uniformed yield (kg ha$^{-1}$) throughout the field, the distribution of bales follows a constant distance pattern easily obtained from the following equation.

$$d_{bales} = \frac{m_{bale} * 10{,}000}{Q_{straw} * w_{headcut}} \qquad [11]$$

where $d_{bales}$ (m) is the travelled distance by a baler since it packs a bale until it packs the next one, $m_{bale}$ (kg) is the mass of one bale, $Q_{straw}$ is the production level of straw (kg ha$^{-1}$) and the $w_{headcut}$ is the working width of balers. From the calculated distance parameter, and taking the starting working point of the baler, it is immediate to determine the coordinates of each bale generated along a field defined by its dimensions and shape.

In order to develop realistic positions from given dimensions of a field, it has been collected historic data regarding wheat (*Triticum aestivum* L.) crops from a set of fields in central Spain as the ones depicted in Fig. 2. Yield from previous seasons together with the main characteristics of machinery and of bales are listed in Table 1. Different capacity constraints possibilities appear depending on the wagon used. There is a wide range of either self-propelled or pull-type bale wagons with different loading capacities depending on the dimension of bales. In Table 1 three different capacities are considered.

However, real production is not uniform throughout a field due to the spatial and temporal variability in the soil-plant-atmosphere system. One of the aims of PA is the generation of maps of yield variation to implement site specific crop management. Fig. 2 represents two fields with strongly different shapes, in both cases it can be clearly appreciated several greenery levels at that vegetative stage. These levels of greenery will correspond at the time of harvest to different yield levels. Therefore, in order to implement a more realistic generator of bales' locations, it has to be considered that yield variability. For that purpose, the problem generator divides the field into different units and each unit is assigned a yield level according to the histogram of frequencies of pixels' green saturation (10 levels of saturation) obtained from masked images from Fig. 3. Yield levels have been assigned ranging from 75% up to 125% the average yield recorded in Table 1.

As an example, let us have a field with a rectangular area of 6.24 ha and dimensions of 120 m × 520 m. Fig. 4(a) shows the uniform distribution of bales on the field, while Fig. 4(b) shows the location of bales

**Table 1.** Significant data on the cultivation of irrigated wheat in Leon province

| | |
|---|---|
| Approximate average yield (kg ha$^{-1}$) | 7,000 |
| Approximate yield of straw (kg ha$^{-1}$) | 3,500 |
| Working width of the cutting head (m) | 6 |
| Mass of the bales (kg bale$^{-1}$) | 700 |
| Number of bales ha$^{-1}$ | 5 |
| Number of different tours (15 bales tour$^{-1}$) | 67 |
| Number of different tours (35 bales tour$^{-1}$) | 30 |
| Number of different tours (108 bales tour$^{-1}$) | 10 |

*Source:* ESYRCE, 2012 (www.magrama.gob.es).

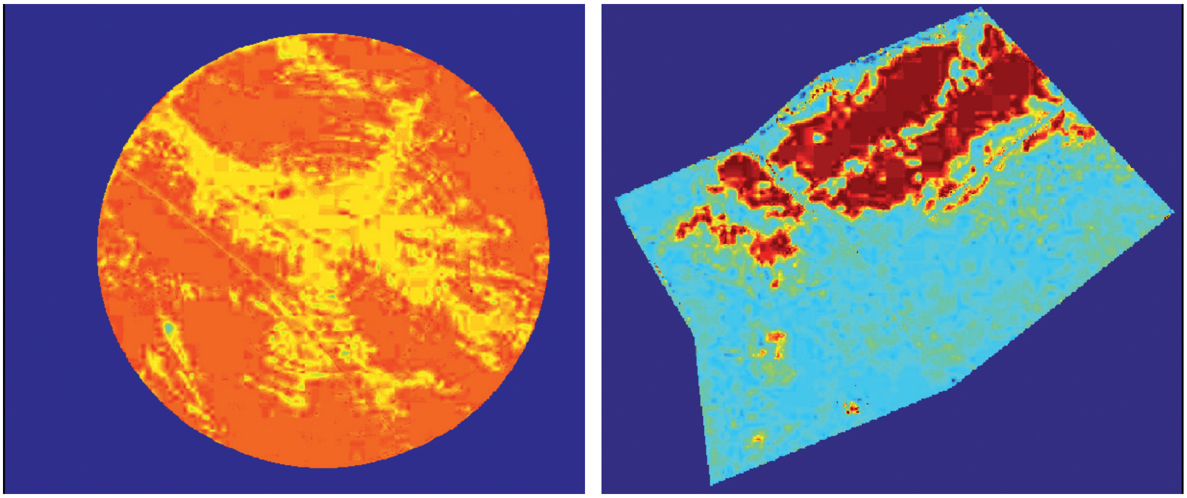

**Figure 3.** Image processing to determine green levels for different shapes.

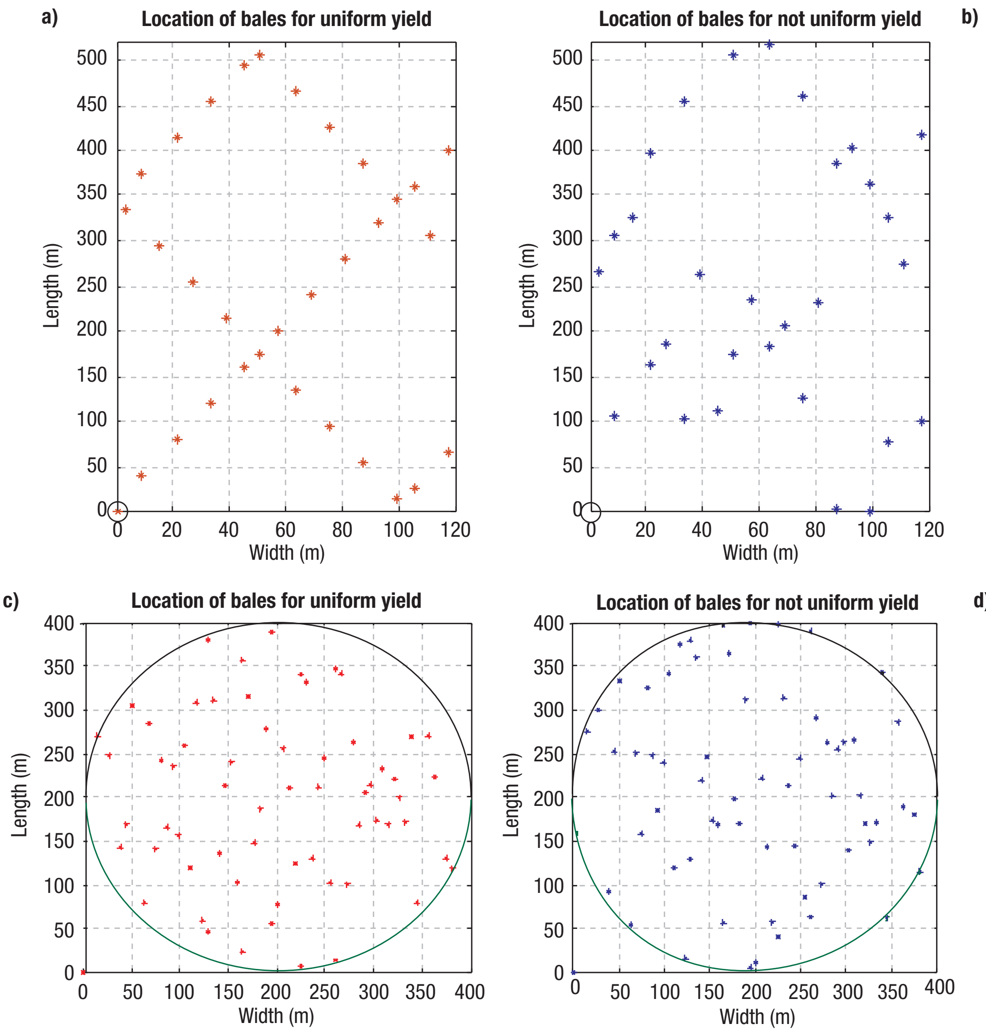


**Figure 4.** Distribution of wheat bales for different shapes: (a) rectangular field and a uniform distribution; (b) rectangular field and yield variability; (c) centre pivot field and a uniform distribution; (d) centre pivot field and yield variability.

when taking into account yield variability. When considering yield variability, any possible location pattern followed throughout the plot disappears. Notice that the entry point to the plot has been circled and it is located at the coordinate's origin. According to the 2010$^{th}$ survey on areas and crop yields in Spain (ESYRCE), published by the Ministry of Agriculture (www.magrama.gob.es), in Central Spain, areas with mechanized sprinkler pivot systems already comprises 26% of the total irrigated area; in wheat corps and for Castilla-León provinces this means 60,000 ha, which represents 4% of total wheat production in the region. Hence, the problem generator has to consider different shaped plots. Fig. 4(c) and Fig. 4(d) show uniform and variable distribution of bales for a centre pivot irrigated plot of 12.56 ha (radius = 200 m) when packed in parallel tracks. Notice that in circle shapes uniform distribution of bales does not follow an easy to see pattern and the effect of yield diversity is not as evident as it was in rectangular plots, which is a consequence of a variable row length at each track.

Taking all previous considerations, a problem generator for rectangular and circular areas has been implemented in commercial software MATLAB® release R2007b. Each instance has the following parameters: {W,L}, width and length of the field; {x0,y0}, location of silo node; M, mass of bales; K, homogeneous capacity of vehicles, $\{Q_{straw}\}$, production per ha of wheat straw; $\{w_{straw}\}$, width of head cuts for forage maize and wheat; U, number of different yield units within the field.

## Results and discussion

The BCP is solved using the HGA approach. Problem instances are generated taking into account the casuistic relative to land division in central Spain as identified in Botey (2009), where it is concluded that 75% of land division is composed of plots between 1 ha and 50 ha. Two different shapes have been considered: rectangular and circular plots. Instances for rectangular plots combine several field dimensions (listed in Table 2) and two different capacities of wagons: 15 and 35 bales/wagon. The silo is located at the origin (0,0); 700 kg is the mass of bales; 3,500 kg ha$^{-1}$ is the yield for wheat straw; 6 m is the width of head cuts; 10 is the number of different yield levels considered within the field; from each instance ten different test problems are generated.

In order to validate the proposed approach, the performance of the proposed HGA is compared to the usual assignment rules that an experienced operator would follow when collecting straw bales. Two different heuristic rules can describe these working patterns: the nearest neighbour heuristic, already described above; and the collection of consecutive bales belonging to adjacent rows, where vehicle movements zigzag among two or three parallel rows. Fig. 5 illustrates both methods for a uniform distribution of bales in the field and a capacity constraint of 6. Left image shows selection between two adjacent rows. Each tour is identified by a different colour; notice that all tours have their starting and ending point at position origin (0,0).

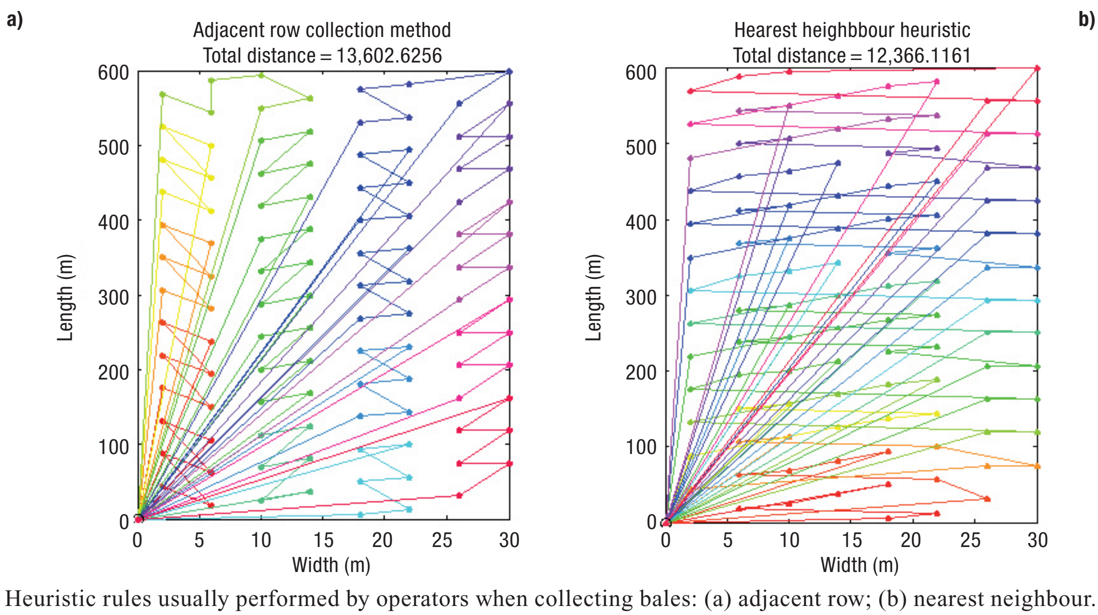


**Figure 5.** Heuristic rules usually performed by operators when collecting bales: (a) adjacent row; (b) nearest neighbour.

**Table 2.** Obtained results for rectangular (a) and circular (b) instances

| | Width (m) | Lenght (m) | Area (ha) | Capacity 35 (bales/wagon) | | | Capacity = 15 (bales/wagon) | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | NNH | HGA | Savings (%) | NNH | HGA | Savings (%) |
| a) *Rectangular instance No.* | | | | | | | | | |
| 1-a | 120 | 600 | 7.2 | 3,261 | 2,080 | 57 | 3,578 | 2,787 | 28 |
| 2-a | 100 | 1,000 | 10.0 | 3,690 | 3,006 | 23 | 6,375 | 4,444 | 43 |
| 3-a | 200 | 505 | 10.1 | 1,885 | 1,300 | 45 | 2,902 | 2,674 | 9 |
| 4-a | 145 | 800 | 11.6 | 4,136 | 3,311 | 25 | 5,582 | 4,930 | 13 |
| 5-a | 296 | 555 | 16.4 | 4,947 | 4,429 | 12 | 6,935 | 6,168 | 12 |
| 6-a | 210 | 800 | 16.8 | 6,131 | 5,053 | 21 | 8,366 | 7,386 | 13 |
| 7-a | 183 | 1,019 | 18.6 | 6,532 | 5,421 | 20 | 11,202 | 9,583 | 17 |
| 8-a | 206 | 1,027 | 21.2 | 7,294 | 6,249 | 17 | 11,400 | 10,610 | 7 |
| 9-a | 410 | 565 | 23.2 | 7,479 | 6,402 | 17 | 10,279 | 9,734 | 6 |
| 10-a | 220 | 1,087 | 23.9 | 8,537 | 7,924 | 8 | 12,403 | 12,011 | 3 |
| 11-a | 430 | 600 | 25.8 | 7,764 | 7,439 | 4 | 11,700 | 11,140 | 5 |
| 12-a | 228 | 1,262 | 28.8 | 12,996 | 10,936 | 19 | 18,671 | 17,845 | 5 |
| 13-a | 416 | 720 | 30.0 | 10,240 | 9,246 | 11 | 13,614 | 13,091 | 4 |
| 14-a | 380 | 1,030 | 39.1 | 14,567 | 13,924 | 5 | 22,551 | 21,576 | 5 |
| 15-a | 572 | 699 | 40.0 | 13,740 | 13,185 | 4 | 21,657 | 21,008 | 3 |
| Average | | | | | | 19 | | | 12 |

| | Radius (m) | Area (ha) | NNH | HGA | Savings (%) | NNH | HGA | Savings (%) |
|---|---|---|---|---|---|---|---|---|
| b) *Circular instance No.* | | | | | | | | |
| 1-b | 150 | 7.1 | 1,840 | 2,935 | 20 | 2,629 | 2,629 | 15 |
| 2-b | 200 | 12.6 | 3,436 | 3,436 | 17 | 5,138 | 5,138 | 17 |
| 3-b | 250 | 19.6 | 5,851 | 4,993 | 17 | 8,807 | 8,030 | 10 |
| 4-b | 300 | 28.3 | 10,205 | 8,551 | 19 | 14,246 | 12,788 | 11 |
| 5-b | 350 | 38.5 | 13,470 | 11,678 | 15 | 20,181 | 18,918 | 7 |
| 6-b | 400 | 50.3 | 18,744 | 16,780 | 12 | 29,081 | 29,081 | 5 |
| Average | | | | | 17 | | | 11 |

NNH: nearest neighbour heuristic. HGA: hybrid genetic algorithm.

Under different working conditions (dispersion of bales, cutting head width, etc) one operator would choose to use one assignment rule against the other due to physical limitations of the collecting process (loss of perspective and local myopia). However the performance of the nearest neighbour heuristic will always have the same or better performance (depending on the number of bales and the number of row) than the adjacent rows heuristic. Nearest neighbour heuristic will be therefore the lower bound to which the HGA approach will be compared.

Table 2 shows the average best results obtained for each instance in terms of total distance travelled. Results are compared to the nearest neighbour heuristic (NNH) procedure and the percentage of savings is calculated. As seen, average percentage of savings varies between 19% and 12% depending on the capacity of the vehicle and on the size and shape of the field. This raises awareness of the appropriateness of implementing this algorithmic approach.

There are almost no previous references to solving the BCP bales in the literature, only Grisso *et al.* (2007) raised a simple instance in which 34 bales scattered over a field should be collected with a vehicle capacity of 6 bales. The performance of the HGA for Grisso's instance after 3,294 iterations improved the solution proposed in Grisso *et al.* (2007) in a around 6%. Fig. 6 shows the solution obtained by the genetic algorithm, each tour is highlighted in colour and as shown all start and end at the origin.

An algorithmic approach based on genetic algorithms and local search heuristics to generate the collec-

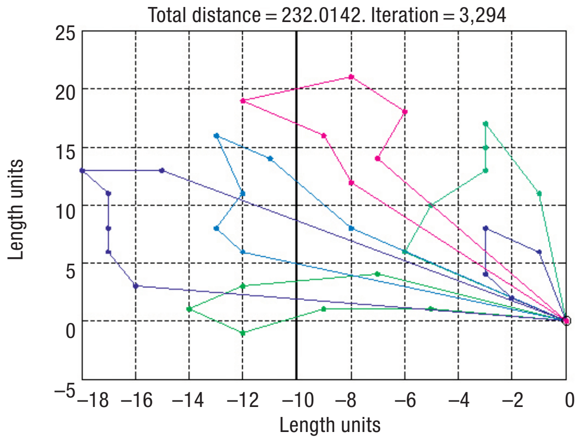


**Figure 6.** Obtained tours by the hybrid genetic algorithm (HGA) for an instance previously proposed in bibliography (Grisso *et al.,* 2007).

ting sequences for loaders and bale wagons has been developed. The resulting sequences are optimal in the sense that they minimise the total travelled distance in the field. The bale collecting problem (BCP) was formulated and modelled in terms of mathematical programming.

Experimental results showed that by using solutions generated by the HGA instead of operator-selected ones, the total distance can be reduced significantly, 15% average in a medium size conventional plot. These savings have been obtained comparing the approach with an experienced operator with no mistakes on his assignment rules. This is way far from real situations so real savings will even be more considerable since sub-optimal patterns may need even corrections which will increase the total distance. This fact was already pointed out when describing adjacent row collecting method, local myopia of operator makes him apply less efficient assignment decisions.

The reduction of travelled distance entails equivalent reduction of fuel consumption and of working time.

Programmable navigation aided systems and auto-steering systems which are already a reality in agricultural machines make possible the appliance of such route planning optimization techniques within an everyday context.

## Acknowledgements

This work was supported in part by the Spanish Government (research project AGL2010-15334).

## References

Amiama C, Bueno J, Ávarez CJ, Pereira JM, 2008. Design and field test of an automatic data acquisition system in a self-propelled forage harvester. Comput Electron Agric 61: 192-200.

Baker BM, Ayechew MA, 2003. A genetic algorithm for the vehicle routing problem. Comput Oper Res 30: 787-800.

Baykasoglu A, Ozbakor L, Tapka P, 2007. Artificial bee colony algorithm and its application to generalized assignment problem. In: Swarm intelligence, focus on ant and particle swarm optimization (Chan FT, ed). I-Tech Education and Publishing, Vienna (Austria). pp: 113-144.

Bentley JJ, 1992. Fast algorithms for geometric traveling salesman problems. ORSA J Comput 4: 387-411.

Bochtis DD, Sorensen CG, 2009. The vehicle routing problem in field logistics part I. Biosystems Eng 104: 447-457.

Bochtis DD, Sorensen CG, 2010. The vehicle routing problem in field logistics: Part II. Biosystems Eng 105: 180-188.

Bochtis DD, Dogoulis P, Busato P, Sorensen CG, Berruto R, Gemtos T, 2013. A flow-shop problem formulation of biomass handling operations scheduling. Comput Electron Agric 91: 49-56.

Botey M, 2009. La concentración parcelaria en Castilla y León. Caracterización de la parcelación a través del análisis multivariante. Doctoral thesis. Univ Politécnica, Madrid, Spain. [In Spanish].

Brady RM, 1985. Optimization strategies gleaned from biological evolution. Nature 317: 804-806.

Chen J, Pan JC, Lin C, 2008. A hybrid genetic algorithm for the re-entrant flow-shop scheduling problem. Expert Syst Appl 34: 570-577.

Cook SE, Bramley RG, 1998. Precision agriculture – Opportunities, benefits and pitfalls. Aust J Exp Agric 38: 753-763.

Cordeau JF, Gendreau M, Laporte G, Potvin JV, Semet F, 2002. A guide to vehicle routing heuristics. J Oper Res Soc 53: 512-522.

Dantzig G, Fulkerson R, Johnson DS, 1954. Solution of a large-scale travelling salesman problem. Oper Res 2: 393-410.

Dasgupta D (ed), 1999. Artificial immune systems and their applications. Springer-Verlag Inc, Berlin, Germany.

Davis L, 1985. Job shop scheduling with genetic algorithms. Proc of the First Int Conf on Genetic Algorithms and their Applications, Pittsburg, PA (USA). July 24-26. pp: 136-140.

De Castro LN, Timmis J, 2002. Artificial immune systems: a new computational approach. Springer-Verlag Inc, London, UK.

Dorigo M, Stützle T, 2004. Ant colony optimization. MIT Press, Cambridge, MA, USA.

Eksioglu B, Vural AV, Reisman A, 2009. The vehicle routing problem: a taxonomic review. Comput Ind Eng 57: 1472-1483.

Garey MR, Johnson DS, 1979. Computers and intractability: a guide to the theory of NP-completeness. WH Freeman & Company, NY.

Gendreau M, Laporte G, Potvin JY, 1998. Metaheuristics for the vehicle routing problem. Technical Report G-98-52, Les Cahiers du GERAD, Montréal, Quebec, Canada.

Gillet BE, Miller LR, 1974. A heuristic algorithm for the vehicle-dispatch problem. Oper Res 22: 340-349.

Goldberg DE, 1989. Genetic algorithms in search, optimization and machine learning. Kluwer Acad Publ, Boston, MA, USA.

Gracia C, Andrés C, Gracia L, 2013. A hybrid approach based on genetic algorithms to solve the problem of cutting structural beams in a metalwork company. J Heuristics 19(2): 253-273.

Grisso RD, Cundiff JS, Vaughan DH, 2007. Investigating machinery management parameters with computers tools, ASABE Conf, Paper 071030.

Hameed IA, Bochtis DD, Sorensen CG, Vougioukas S, 2012. An object-oriented model for simulating agricultural i-field machinery activities. Comput Electron Agric 81: 24-32.

Holland JH, 1975. Adaptation in natural and artificial systems (Holland JH, ed). Ann Arbor MI Univ of Michigan Press, MI, USA.

Jünger M, Reinelt G, Rinaldi G, 1995. The traveling salesman problem. In: Network models. Handbooks on Operations Research and Management Science 7 (Ball MO, Magnanti TL, Monma CL, Nemhauser GL, eds). Elsevier, Amsterdam, pp: 225-330.

Kennedy JF, Kennedy J, Eberhart R, Shi Y, 2001. Swarm intelligence. Academic Press Inc, London.

Laporte G, Gendreau M, Potvin JY, 2000. Classical and modern heuristics for the vehicle routing problem. Int Trans Oper Res 7: 285-300.

Martin O, Otto SW, Felten EW, 1991. Large-step markov chains for the travelling salesman problem. Complex Syst 5(3): 299-326.

Nikkilä R, Seilonen I, Koskinen K, 2010. Software architecture for farm management information systems in precision agriculture. Comput Electron Agric 70: 328-336.

Sorensen CG, Pesonen L, Bochtis DD, Vougioukas SG, Suomi P, 2011. Functional requirements for a future farm management information system. Comput Electron Agric 76: 266-276.

Toth P, Vigo D, 2002. Branch-and-bound algorithms for the capacitated vehicle routing problem. In: The vehicle routing problem (Toth P, Vigo D, eds). SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA (USA). pp: 29-51.

Wang C, Lu J, 2010. An effective evolutionary algorithm for the practical capacitated vehicle routing problems. J Intell Manuf 2: 363-375.

Zhang N, Wang M, Wang N, 2002. Precision agriculture – A worldwide overview, Comput Electron Agric 36(2-3): 113-132.