IEMS Vol. 1, No. 1, pp. 10-18, December 2002.

Efficiently Solving Dispatching Process Problems in Nurseries by Heuristic Techniques

Erhan Kozan†
School of Mathematical Sciences
Queensland University of Technology, GPO Box 2434 Brisbane Qld 4001, AUSTRALIA
Tel: +617-3864-1029, Fax: +617-3864-2310, E-mail: e.kozan@qut.edu.au

Abstract. A more comprehensive analytical framework for examining the relative merits of alternative dispatching process policies for nurseries is developed in this paper. The efficiency of the dispatch process of plants in a nursery is analysed using a vehicle routing model. The problem then involves determining in what order each vehicle should visit its locations. The problem is NP-hard. Several heuristic techniques are used to solve a real life nursery sequencing problem. The results obtained by these heuristic techniques are compared with each other and the current sequencing of orders. The model with some minor alterations can be also used to minimise the dispatching and collecting process in different agricultural plants.

Keywords: heuristics; vehicle routing; dispatching; nursery.

1. INTRODUCTION

This paper outlines the development of a model to analyse the transporting of plants from the growing area to the dispatch shed for detailing and packaging, and the logistics of such operations. The dispatch process of plants from production nurseries is the most costly, labour intensive and inefficient operation in the Australian nursery industry. The dispatch process is defined as all the tasks which are performed between the time an order is received to the time the plant is awaiting loading to external transport.

The main factors affecting a nursery dispatching can be detailed as follows:

- size of the nursery;
- number and size of orders;
- number of plant species in each of the three areas (that is open, shaded and greenhouse) and number of locations available in each of the these areas;
- number of growing bays required and monthly demands by each species;
- location of the plants in relation to the dispatch shed;
- number of periods;
- tasks involved and sequence of plant processing;
- number and type of plants processed and pot size used for these plants;
- number of persons involved in tasks and their duration;
- unproductive time;
- type of equipment available for transportation of plants; and
- number of times it is necessary to go to a specific location of the nursery.

The objective is to schedule a series of routes such that the minimum number of vehicles is used and the total distance is minimised with all locations being serviced. The problem then involves determining in what order each vehicle should visit its locations.

There is one route per vehicle, which starts and finishes at a central facility area. The problem is NP-hard, where NP stands for non-deterministic polynomial . There is currently no known polynomial time algorithm for computing the optimal solution, however, any correct solution can be verified in polynomial time with respect to the size of the problem (Christofides *et al.*, 1989 and Lenstra J. and Kan R. 1981). The classical text on computational complexity is by Garry and Johnson (1979) and Parker and Rardin (1988).

There are several possible objective functions for the problem. These include minimising distance, minimising travelling time, minimising the number of vehicles and minimising total cost. The basic Vehicle Routing Problem (VRP) ignores a large number and variety of additional constraints and extensions that are often found in real-life problems. There are many extensions of the basic VRP to account for the large number of practical applications (Christofides *et al.*, 1989). There are many derivatives of the VRP, which are usually of the form of extra constraints or mixtures of constraints. Examples of these can be found in Kozan (2000), Dror *et al.* (1994), Balakrishnan (1993) and Laporte *et al.* (1992). Further constraints that can be included are as follows: duration of the route; start and end locations for the route/driver; start times of routes; weight and volumes restrictions on vehicle load; loading constraints and restrictions; rules for split deliveries (pickups) if any; and multiple routes per vehicle. Research into solutions for the VRP have been two pronged, optimal and near optimal (heuristic) solutions. Exact solution techniques used to solve the VRP can be put into three broad categories; tree searches; dynamic programming; and integer linear programming. Details of these techniques can be found in Christofides (1989) and Laporte (1992). Kulkarni and Brave (1985) provided integer programming formulations of vehicle routing problems. They introduced several formulations for the travelling salesman problem, the m-travelling salesman problem, the vehicle routing problem and the multi-depot vehicle routing problem. Desrochers *et al.* (1988) improved the Kulkarni and Brave (1985) model with additional extensions for time horizon planning.

The computational complexity of NP-Hard problems increases exponentially with the number of components in the schedule. This makes it difficult to solve in reasonable time with the current exact solution techniques, (ie. Branch and Bound, Tree Searches). This implies that for large size real life problems heuristic techniques have to be used. Heuristic techniques can be grouped into the following classifications Osman (1993): constructive methods; two-step methods; exact but incomplete tree search methods; and improvement methods. Constructive methods are those that build up vehicle tours by inserting at each step a location according to some savings measure until all locations are served. The most used of these is the Clark and Wright (1964) Savings Method. Bramel & Simchi-Levi (1995) introduced a general framework for modelling routing problem based on formulating them as a traditional location problem called the capacitated concentrator location problem. They applied this method to two classical routing problems: the capacitated vehicle routing problem and the inventory routing problem.

Other two-step algorithms can be found in Christofides (1989) and Laporte *et al.* (1992). Improvement methods iteratively improve a given solution by making local changes. Osman (1993) made the observation that most iterative improvement methods start by using a constructive method to obtain an initial feasible solution, and used an improvement technique that reduced the cost of the tour by making the local changes while maintaining feasibility. More popular improvement methods are simulated annealing, genetic algorithms and tabu searches. These heuristic techniques have been used to solve the real life material-handling problem, which yields a good solution to a problem, but cannot be guaranteed to produce an optimum.

Kirkpatrick *et al.* (1993) discovered the analogy between achieving a low energy state in a metal and optimisation problem. A metal allowed to cool slowly from a high temperature to a low temperature will reach a state of minimum energy. With this in mind, Kirkpatrick developed simulated annealing to solve the travelling salesman problems using the Metropolis algorithm.

Holland (1998) devised a new research mechanism, which he called genetic algorithm, based on Darwin's (1859) principle of natural selection. In its simple form, genetic algorithm recursively applies the concepts of selection, crossover and mutation to a randomly generated population of promising solutions with the best solution found being reported.

2. NURSERY DISPATCHING MODEL

In relation to the nursery sequencing problem the central facility of the depot is the dispatch shed. All trailers leave and return to the dispatch shed. The objective is to minimise the travelling distance when collecting orders. The orders are available at the beginning of each morning and consequently the demands for each species are known in advance. In the case of using more than one trailer their capacities are assumed homogenous. Capacity is measured in terms of the number of 140mm pots that can be transported by a trailer. A conversion factor is used to convert all demands in terms of 140mm pots. Travelling distances between all locations are measured for an input to the model. If the demand for a plant species exceeds the trailer capacity the load is split into two (or more) full trailer loads and only the remainder considered.

Notations

dem_i : Demand for plant species i.

$dist_{ij}$: The total distance to travel from plant species i to j. Where $i = 2, 3, ..., n$. ($i = 1$ refers to the dispatch shed) and $j = 2, 3, ..., n$.

c : The capacity of a trailer in terms of 140mm pots.

u_i and u_j : Arbitrary real numbers

$$x_{ijr} = \begin{cases} 1 & \text{if in route } r \text{ the location of plant } j \text{ is visited just} \\ & \text{after the location of plant } i. \\ 0 & \text{otherwise} \end{cases}$$

where $r = 1, 2, \cdots, m$.

$$y_{ir} = \begin{cases} 1, & \text{if plant species } i \text{ is visited in route } r \\ 0, & \text{otherwise} \end{cases}$$

Objective

The objective is to minimise the total travelling time.

$$\text{Minimise } Z = \sum_i \sum_j \sum_r dist_{ij} x_{ijr} \quad (1)$$

Constraints

The model is subject to the following constraints:

$$\sum_r y_{ir} = \begin{cases} 1 & i = 2, 3, \cdots, n \\ m & i = 1 \end{cases} \quad (2)$$

Constraint (2) ensures that every plant species is allocated to some route (except the dispatch shed which is visited in every route).

$$\sum_i dem_i y_{ir} \leq c \quad \forall r \quad (3)$$

Constraint (3) is the vehicle capacity constraint. The number of pots collected in any one route must not exceed the capacity of the trailer.

$$\sum_j x_{ijr} = \sum_i x_{jir} = y_{ir} \quad \forall i, r \quad (4)$$

Constraint (4) ensures that if a plant location is visited in a particular route it must also be left in that same route.

$$\sum_{i,j,\in S} x_{ijr} \leq |S| - 1 \ \forall S \subseteq \{2, 3, \cdots, n\} \text{ and } \forall r \quad (5)$$

Constraint (5) prohibits any subtours ensuring that whole tours are completed at all times.

$$y_{ir} \in \{0, 1\} \quad \forall i, r \quad (6)$$

$$x_{ijr} \in \{0, 1\} \quad \forall i, j, r \quad (7)$$

Assumptions

The following assumptions are made to simplify the situation:

- At the beginning of each day the orders are sorted and the total number of each species required is known. The different species are divided into groups by location and the number of plants required in that specific location recorded;
- There exists uniform stock and the selection of plants is not required. This implies that plants are picked one after each other and no sorting through plants is required;
- The distances from the dispatch area to each growing bay are define as the distance travelled from the dispatch area to the midpoint of the front of a growing bay plus the vertical distance up to the middle of that growing bay;
- All routes are either horizontal or vertical;
- The nursery size is assumed large enough so that there is generally enough of each species to fill at least one trailer or perhaps make allowances for those species in low demand and group these together;
- A conversion factor is set to convert the number of pots of each of the six sizes held by a trailer to 140mm pots;
- It is assumed that regardless of the picking strategy plants should be allocated according to demand. That is, having plants in higher demand closer to the dispatch shed, consequently reduce the distance travelled to collect orders;
- Once a species is allocated a location in the nursery it stays there until demanded.

3. SOLUTION TECHNIQUES

A vehicle routing problem has been formulated as mixed Integer programming to solve the problem of collecting orders in an efficient manner and solved for a smaller size problem using the Generalised Algebraic Modelling System, GAMS (1998). An optimal solution for a real life problem can not be found by packages like GAMS in a reasonable time period because it would be impractical to wait several hours for a solution when pickers have to commence collecting orders early in the morning. So heuristic techniques have been used to solve the real life nursery sequencing problem which yields a good solution to a problem, but cannot be guaranteed to produce an optimum. Three heuristic techniques, namely Clark and Wright method, sweep algorithm and genetic algorithm have been applied on a real data set. The heuristic techniques are validated by comparing the results for a number of small problems with the present sequencing of orders. The paper presents the results of applying the model to a problem that is indicative of the size experienced in the nursery industry, with three trailers and 22 locations. The three heuristic techniques are then compared for the solution values obtained and the time frame required.

3.1 Clark and Wright algorithm

The Clark and Wright algorithm initially constructs routes from the depot and each of the locations ($n-1$ routes). Then measures of savings are determined by calculating the amount of time/distance saved by linking two locations. The algorithm then joins the pairs in decreasing distance value of savings, subject to constraints (Equations (1) to (7)). Two algorithms are coded, one for the parallel version and the other for the sequential version.

3.2 Sweep Algorithm

The sweep algorithm method uses both distances from every location to every other location, and the angle of rotation from the dispatch shed. The method sweeps from a location in order of angle of rotation while the capacity constraints are not violated. The locations included in the sweep become the next cluster. The method continues sweeping until all locations are assigned to a cluster. Once the clusters are assigned, a Travelling Salesman Problem (TSP) is solved for each cluster to arrive at the minimum route distance for that cluster.

In contrast to the Clarke-Wright algorithm, the sweep algorithm requires geographical co-ordinates for each location. There are several variations of the sweep heuristic. The following steps listed below give an overview of the sweep algorithm process as implemented.

Step 1. Pass an arc through the depot in a northerly direction. The arc is sweep in a clockwise direction with the depot as the pivot. A record is kept of the order of the customers.

Step 2. Choose an unused vehicle.

Step 3. Starting with the first unrouted customer on the list, include consecutive customers until the capacity constraint of the vehicle is reached.

Step 4. If all customers are swept' or if all vehicles have been used go to Step 5, else return to Step 2.

Step 5. Solve the TSP for every set of customers assigned to a vehicle to form the final routes. The actual sequence can be determined as a travelling salesman problem since there is no longer a capacity constraint.

Step 6. Carry out steps 1 to 5 for all possible starting points, that is starting from each of the different directions (north, south, east and west). The whole procedure is again repeated but this time sweeping in an anticlockwise direction. The best of the eight solutions is chosen.

The sweep heuristic produces several solutions which can often be advantageous especially when there are other constraints to consider.

3.3 Genetic Algorithm

Genetic Algorithms (GA) are a subset of what is known as Evolutionary Algorithms. Other major subsets are Evolutionary Programs, Evolutionary Strategies, Classifier Systems and Genetic Programming. A GA is an adaptive search technique, based on the principles and mechanisms of natural selection and survival of the fitness from natural evolution. GAs grew out of Holland's (1998) study of adaptation in artificial and natural systems. In this study, GA algorithms had been used because it is generally accepted that by simulating natural evolution in this way, a GA can effectively search a very large broad class of problems' domains and easily solve complex problems. Relatively good results that have been reported with even the simplest GA implementation. In addition, by emulating biological selection and reproduction techniques, a GA can perform searches in an independent manner. If the genetic representation of two different vehicle routing problems is identical, then it is usually possible to optimise both problems with few changes to the underlying GA. In this case only changing the operational parameters of the GA is enough.

Goldberg (1989) defined GA as search procedures based on the mechanics of natural selection and natural genetics. GA represent potential solutions to a problem as genotypes. These genotypes (chromosomes) form a population, which undergo processes that resemble natural genetics.

A genetic algorithm (as any evolution program) for a particular problem must have the following five components:

- a genetic representation for potential solutions to the problem;
- a way to create an initial population of potential solutions;
- an evaluation function that plays the role of the environment, rating solutions in terms of their fitness ;
- genetic operators that alter the composition of children; and
- values for various parameters that the genetic algorithm uses (population size, probabilities of applying genetic operators, etc.).

The genetic operators referred to above are generally of two types: crossover and mutations. The crossover operator is the method of transforming a pair of surviving genotypes into a pair of offspring genotypes. The classical crossover involves cutting each genotype into two segments and swapping the segments, creating two different genotypes with characteristics from each parent. This carrying of segments allows the possibility of good string segments to be preserved. Each genotype will have a probability of mutation. "Mutation arbitrarily alters one or more genes of a selected chromosome, by a random change with a probability equal to the mutation rate" (Michalewicz, 1994).

Typically, a binary encoding to map a GA chromosome to a single point in the problem space is used. In many functional problems an n-dimensional problem must be solved. To do this, a parameter set is used and it is partitioned into bit strings. Each bit-string encodes a single parameter. The classic operators, which act on these bit-strings, are crossovers and mutations. Crossovers typically involve exchanging randomly selected bit-string chunks between two parents to create two children genotypes. Mutation involves scanning each bit in the bit-string, and with low probability, toggling that bit. For this implementation a path representation is used because the path representation is perhaps the most natural representation of a tour. In this representation the chromosomes are a string, of length equal to the number of locations (racks), of integers with each number occurring only once. That is, each chromosome contains all integers from one to the number of locations. The order of the genes in a chromosome represents the path of the tour. If in the following, each number represents a particular rack then a chromosome representation (3 2 1 5 4) is a tour from rack represented by 3 to the rack represented by 2 etc.

Many methods are available in GA literature but most of them are only suitable for binary coding. Since the routing problem is a permutation problem the choice of crossover method is limited. Partially mapped crossover, cycle crossover and order crossover methods are the most suitable of these for vehicle routing problems. Each of these operators creates a child from two parents and qualities are transferred from both parents to the child. Partially mapped crossover method builds an offspring by choosing a subsequence of a tour from one parent and preserving the order and position of as many locations as possible from the other parents, see Goldberg (1989). The order crossover method builds offspring by choosing a subsequence of a tour from one parent and preserving the relative order of the locations from the other parent, see Davis (1991) for details. The cycle crossover method builds offspring in such a way that each location (and its position) comes from one of the parents and preserves the absolute position of the elements in the parent sequence, (see Oliver *et al.*(1987) for details). The more popular partially mapped crossover method is used in this study and explained in detail at Step 8 below.

The following steps give an overview of the genetic algorithm process as implemented.

Step 1. Initialise the GA parameters of potential solutions. Set the maximum number of generations, the maximum number of chromosomes in a population, population size, population renewal rate, probability of crossover and probability of mutation. The coding which is used is not a binary coding, because the problem is a permutation problem and the output is a permutation of the input. Randomly allocate the numbers 1 to the total number of locations N for each chromosome.

Repeat (Steps 2-9) for pre-defined iterations.

Step 2. Calculate the travelling time between gene i and $i+1$ and picking time at $i+1$ for each gene in the chromosome. For the selection process (selection of a new population with respect to the probability distribution based on fitness values) a roulette wheel method is used. It assigns a probability to each chromosome i, computed as the proportion, $p_i = p_i / \sum_i p_i$ where p_i is the fitness of chromosome i. A parent is then randomly selected based on this probability

Step 3. Calculate tour time for each chromosome (summation of times calculated in Step 2). The solution is then checked to satisfy the physical constraints. The chromosomes that don't satisfy the constraints are repaired in the initialisation of the first generation. If in generation 1 all constraints are satisfied then algorithms ensures that in the future generations this will be true.

Step 4. Compare all tour times with global minimum, and replace the global with the new minimum tour if necessary. If all the constraints are satisfied the fitness function evaluate using objective function, is calculated. The tour time acts as the evaluation function for the fitness of each chromosome.

Step 5. Create probabilities of survival. This gives the probability that a chromosome will survive to be used as a parent to produce offspring for the next generation of the genetic algorithm. The probability is inversely proportional to the tour time of the chromosomes, and is calculated by dividing the tour time of the minimum chromosome by the time of the chromosome in question, then dividing this by the sum of these divisions. This gives probabilities summing to one, where chromosomes with small tour times have greater chance of surviving.

Step 6. Use the probabilities from Step 5 to randomly generate the surviving population. Note that chromosomes with high probability of survival can actually increase in numbers causing multiple copies of the chromosome.

Step 7. For the population of surviving chromosomes, pairs are chosen randomly to undergo the crossover operator. The random allocation helps to vary the potential offspring.

Step 8. Crossover is the method that mixes the genes of two parents to obtain off spring. In Partially mapped crossover, firstly two crosspoints are selected randomly between 0 and N (the length of gene). Genes from the first parent that fall between the two crosspoints are copied into the same positions of the offspring. The remaining order is determined by the second parent. Non duplicated genes are copied from the second parent to the offspring beginning at the position follows the second crosspoint. Both the second parent and the offsprings are traversed circularly from that point. The surrounding positions are filled from the other parent, keeping where possible the same position and order. A copy of the parents next non duplicative genes is placed in the next available child position. An example of how the crossover works in a data set used is shown below. If cut points are chosen at 3 and 17 (indicated by |) then examples of two parents could be

P1 = (1 10 4 | 5 2 13 17 16 18 21 19 14 12 15 20 6 9 | 3 7 8 11)

P2 = (8 3 11 | 4 6 18 21 20 13 14 19 17 12 16 15 9 1 | 5 7 10 2)

Then the two offspring would look as follows ('x' represents still to be determined).

O1 = (x x x | 4 6 18 21 20 13 14 19 17 12 16 15 9 1 | x x x x)

O2 = (x x x | 5 2 13 17 16 18 21 19 14 12 15 20 6 9 | x x x x)

The next step is to place locations in offsprings from original parent, provided no conflict is encountered.

O1 = (x 10 x | 4 6 18 21 20 13 14 19 17 12 16 15 9 1 | 3 7 8 11)
O2 = (8 3 11 | 5 2 13 17 16 18 21 19 14 12 15 20 6 9 | x 7 10 x)

The other values are gained by placing the opposite location to that which is causing the conflict. For example the third position of offspring number 1 would have been 4 except that 4 is now at position 4, so position three is replaced with location 5. Other replacements are more complicated and involve following trail until a free value that does not occur exists. An example of this is position one offspring one, this should have been replaced with 9, but, 9 also exists in segment so we choose 6, but, 6 is also in segment so we choose 2. This rule results in the following two offspring:

O1 = (2 10 5 |4 6 18 21 20 13 14 19 17 12 16 15 9 1 | 3 7 8 11)
O2 = (8 3 11 |5 2 13 17 16 18 21 19 14 12 15 20 6 9 | 4 7 10 1)

Step 9. Randomly mutate genes with mutation rate probability. If mutation occurs (only one per chromosome is allowed), two locations will swap in position within the chromosome.
End

4. RESULTS

The above heuristics have been applied to a given day's orders. A record of a day's orders along with the actual sequence of collection, a map of the stopping places and distances were available.

Different species often require different pot sizes. These pot sizes usually range from 100mm to 350mm. It is for this reason that it has become standard practice within the nurseries to convert all pot sizes to a base unit, this being a 140mm pot. That is, all demands have been converted into the equivalent number of 140mm pots and provided in Table 1. The capacity of a trailer is 360 pots, in terms of the number of 140mm pots that it can transport at any one time.

The stopping numbers is provided in Table 2. These numbers show the actual order in which the plants are collected. There are 21 different locations and the distances between each location have been measured and used in the model. At any one location several different plant species are loaded onto the trailer.

The Clarke-Wright savings algorithm has been

Table 1. Pot conversion factors

Pot size	100	125	140*	175	200	300	350
Factor	0.51	0.80	1.00	1.55	2.04	5.00	13.00

* *base unit*

Table 2. Present dispatching process

Stop No	Species and Pot Size		# of Pots	Conversion Factor	Converted # of Pots
Trailer 1					
1	Dipladenia	140	6	1.00	6.00
2	Agapanthus	200	12	2.04	25.00
3	Bushy Blue	200	31	2.04	63.24
	Gretel	350	2	13.00	26.00
	Primrose	350	4	13.00	52.00
	Bushy Blue	140	32	1.00	92.00
4	Annabel	200	5	2.04	10.20
	Annabel	350	2	13.00	26.00
5	Gretel	200	5	2.04	10.20
	Carmella	200	5	2.04	10.20
	Celia	200	5	2.04	10.20
				Total	333.00
Trailer 2					
6	Golden beauty	140	36	1.00	36.00
	Pink Numenos	140	12	1.00	12.00
	Evolvulus	140	30	1.00	30.00
7	Prima Dona	140	6	1.00	6.00
8	Liriope	140	144	1.00	144.00
9	Vibernum	140	12	1.00	12.00
10	Bushy Blue	300	3	5.00	15.00
	Ballerina	300	9	5.00	45.00
	Duranta	300	1	5.00	5.00
11	Riding Hood	200	12	2.04	24.48
	My Fair Lady	200	2	2.04	4.08
12	Scarlet Pimpernal	200	13	2.04	26.00
				Total	359.00
Trailer 3					
13	Dracellia	100	10	0.51	5.10
	Dryopeteris	100	5	0.51	2.55
	Humata	100	5	0.51	2.55
	Aglamorpha	100	5	0.51	2.55
14	Spathiphyllum	140	6	1.00	6.00
	Sandra	140	6	1.00	6.00
	Sandra	175	3	1.55	4.65
	Emeraldbeauty	140	12	1.00	12.00
15	Gretel	140	6	1.00	6.00
16	Primrose	200	1	2.04	3.00
17	Springfire	300	4	5.00	20.00
18	Victoria	175	27	1.55	42.00
19	Victoria	200	4	2.04	9.00
20	Swan Lake	140	6	1.00	6.00
	Majestic	200	4	2.04	8.16
	Misty Pink	200	4	2.04	8.16
	Golden Yulow	200	10	2.04	20.4
	Pink Parpait	200	10	2.04	20.4
21	Merlin's Magic	200	11	2.04	23.00
				Total	209.00

Table 3. Results from the Clarke-Wright algorithm

Parallel Version		
Trailer	Best tour	
1	Disp-9-5-16-4-3- 2-1-15-Disp*	
2	Disp-10-17-8-7-6-13-14-Disp	
3	Disp-12-21-20-19-18-11-Disp	
	Total travelling distance (meters)	2032
Sequential Version		
Trailer	Best tour	
1	Disp-15-1-2-3-4-16-5-7-9-DISP	
2	DISP-10-17-8-6-18-19-DISP	
3	DISP-11-20-21-12-14-13-DISP	
	Total travelling distance (meters)	2056

* *Sequence of Picking from location*

solved both sequentially and in parallel and the results are provided in Table 3. While the solutions above do give the order of visiting the plant species it is still beneficial to solve the travelling salesman problem for each trailer in the final allocation to obtain the true optimum order of visiting within each subset. Some improvements were found when each subset was solved using the TSP. Table 4 shows the sweep algorithm results for sweeping in a clockwise direction and anti-clockwise direction for initial

Table 4. Sweeping algorithm's results

Sweeping in a Clockwise Direction		
Trailer	Best tour using TSP	
1	DISP-17-2-3-1-15-14-13-DISP	
2	DISP-12-6-7-8-4-10-DISP	
3	DISP-20-21-9-16-5-19-18-11-DISP	
	Total travelling distance (meters)	2155
Sweeping in an anti-clockwise direction		
Trailer	Best tour using TSP	
1	DISP-12-21-20-9-16-5-19-18-11-DISP	
2	DISP-10-4-8-7-6-17-DISP	
3	DISP-13-14-15-1-3-2-DISP	
	Total travelling distance (meters)	2146

Table 5. Genetic algorithm

Trailer	Best tour using GA	
1	DISP- 13-14- 12- 21-20- 11- DISP	
2	DISP-15-1-2-3-4-19-18-DISP	
3	DISP-10-17-6-7-8-5-16-9-DISP	
	Total travelling distance (meters)	1943

starting direction West. Genetic algorithm results are given in Table 5. It is clear that for this particular set of data the GA provides the best solution. This solution results in 20.3% savings in travelling distance.

The best total travelling distance of the trailers have been determined by GA and shown on the layout of nursery in Figure 1.

It is interesting to note that even after 3.6 million iterations in 1960 minutes using branch and bound the solution is not as good as the worst performing heuristic technique. The optimum solution could not be found because of computer memory limitations and time restrictions. The results from each of the heuristic techniques have been compared in terms of the solution value and the time required in obtaining these results. The results of all the heuristic techniques are summarised in Table 6. Set up time for the genetic algorithm is the longest, about ten minutes. The set up times of the other heuristcs is very close to each other, about five minutes. Therefore, obtaining a solution by genetic algorithm takes about ten minutes longer than the other methods. However, this time difference is negligible because within the time window specified by the nursery manger, the method gives a 20.37% travelling distance saving.

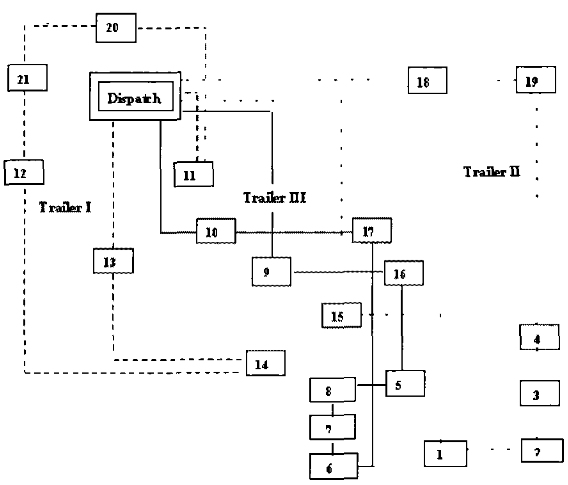

Figure 1. Presentation of best tours on the layout of nursery

Table 6. Comparison of techniques

Techniques	Travelling distance as meters	Running Time as minutes	Saving as %
Benchmark actual distance	2440		0.00
Clarke-Wright savings algorithm-parallel	2032	< 1	16.72
Clarke-Wright savings algorithm sequential	2056	< 1	15.74
Sweep algorithm in a clockwise direction	2155	< 1	11.68
Sweep algorithm in an anti-clockwise direction	2146	< 1	12.01
Branch and bound (360000 iterations)	2192	1960	10.16
Genetic algorithm	1943	4.8	20.37

5. CONCLUSIONS AND FUTURE WORK

It has been shown that heuristic techniques can reduce the travelling distance substantially. From the results above it can be seen that significant savings can be achieved by implementing a method of collecting daily orders and an improved nursery plant layout. The collecting of orders is a task performed on a daily basis and is by far the area which has the most potential for reducing costs.

There are some practical considerations to address before a heuristic technique can be implemented. After determining the routes for the different trailers another question arises: In what order should the trailer loads be collected?' The order will make no difference to the total travelling distance but it will effect the number of trolleys waiting in the dispatch shed.

A quick turnover or species which must be visited regularly for spraying, pruning, or trimming, should be located as close as possible to the operational areas (potting/propagation, dispatch). Species which have a slow turnover or low maintenance should occupy the furthermost reaches of the nursery. So, before applying these heuristics, an optimal plant layout of the nursery should be obtained according to yearly demands

The placement of operational facilities within the nursery (ie. dispatch, potting/propagation areas) can have a large influence on the total distance walked by nursery workers, or the distance a product is carted, over a given time period.

If the consolidated plant pull sheet is prepared according to orders, consecutive orders collated until a trailer-load of plants is totalled for a pull sheet, then this may result in a substantial degree of retracing steps during pick-ups. Similarly, in the dispatch shed, if numerous plants of one species are detailed together there is less stop-start' time compared with detailing order by order. These considerations are particularly important in large nurseries where travelling distances are much more critical than in small nurseries. Then optimum order of plant species collected by trailers will be determined to minimise the distances travelled.

When a trailer load is completed it returns to the dispatch shed where it is unloaded and the preparation of individual orders is started. Incomplete orders are stored on trolleys in the dispatch shed. For example, there are five plant species required in a particular order. Three species may be collected in the first trailer load while the other two on the last. This will consequently result in the trolley designated to this order waiting until the last trailer load is completed. Having worked out the routes for each trailer load the idea is to now order the trailer loads to minimise the number of incomplete orders. Ideally it is desirable to have complete orders started and finished on the same trailer but this is often impossible to achieve.

As a result of this research it can be concluded that Australian production nurseries need a good vehicle routing and an efficient nursery layout. It is important to note that a changed plant layout will in turn effect the collection of orders and of course the distances used to determine the order of collecting plants. Therefore, firstly the nursery layout should be optimised and then the vehicle routing system should be implemented.

ACKNOWLEDGMENT

This research was partly founded by Department of Primary Industries, Queensland, Australia. I would like to thank research assistant Joan Berger for helping in this research project.

REFERENCES

Balakrishnan, N. (1993) Simple Heuristics for the Vehicle Routing Problem with Soft Time Windows. *Journal of the Operational Research Society*, **44**, 279-287.

Bramel, J. and Simchi-Levi, D. (1995) A Location Based Heuristic for General Routing Problems. *Operation Research*, **43**, 649-660.

Christofides, N., Mingozzi, A. and Toth, P. (1989) The Vehicle Routing Problem. In *Combinatorial Optimization*, Christofides N., Mingozzi A., Toth P. and Sandi C. (eds.), John Wiley, New York, 315-338.

Clark, G. and Wright, J. W. (1964) Scheduling of Vehicle from a Central Depot to a Number of Delivery Points. *Operations Research*, **12**, 568-581.

Darwin, C. (1859) The Origins of Species by Means of Natural Selection. London, Penguin Classics.

Davis, L. (1991) *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.

Desrochers, M., Lenstra, J. K., Savelsbergh, M. W. P. and Soumis, F. (1988) Vehicle Routing with Time Windows: Optimization and Approximation, in *Vehicle Routing: Methods and Studies*, Studies in Management Science and Systems. Elsevier Science Publishers B.V. Amsterdam, 65-84.

Dror, M., Laporte, G. and Trudeau, P. (1994) Vehicle Routing with Split Deliveries. *Discrete Applied Mathematics*, **50**, 239-254.

GAMS-2.50A. (1998) GAMS *Development Corporation* 1217 Potomac Street NW Washington.

Garry, M. R. and Johnson, D. S. (1979) *Computers and Intractability: A guide to the Theory of NP-Completenes*. W. H. Freeman and Company: San Fransisco.

Goldberg, D. E. (1989) *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, Reading, MA.

Holland, J. (1998) Adaptation in Natural and Artificial Systems. MIT Press.

Kirkpatrick, S., Gelatt, C. D. (JR) and Vecchi, M. P. (1993), Optimising by Simulated Annealing, *Science*, **220.4598**, 671-680.

Kozan, E. (2000) *An Integrated Material Handling System for a Truck Assembly Plant*, Journal of the Operational Research Society, **51.9**, 263-271.

Kulkarni, R. V. and Brave, P. R. (1985) Integer programming formulations of vehicle routing problems. *European Journal of Operational Research*, **20**, 58-67.

Laporte, G. (1992) The Vehicle Routing Problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, **59**, 345-358.

Laporte, G., Mercure, H. and Nobert, Y. (1992) A Branch and Bound Algorithm for a class of Asymmetrical Vehicle Routing Problem. *Journal of the Operational Research Society*, **43**, 469-481.

Lenstra, J. and Kan, R. (1981) Complexity of the Vehicle Routing and Scheduling Problems. *Network*, **11**(2), 221-228.

Michalewicz, Z. (1994) *Genetic Algorithms +Data Structures = Evolution Programs*. Springer-Verlag, New York.

Oliver, I. M., Smith, D. J. and Holland, J. R. C. (1987) A Study of Permutation Crossover Operators on the Travelling Salesman Problem In Proceedings of the Second International Conference on Genetic Algorithms, Grefenstette JJ (ed), Lawrance Erlbaum Associates, Hillsdale, NJ, 159-166.

Osman, I. H. (1993) Metastrategy, Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem, in Tabu Search. *Annals of Operations Research*, **41**, 421-451.

Parker, R. G. and Rardin, R. L. (1988) *Discrete Optimisation.* Academic Press. San Diego.