Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 177 (2007) 1751–1763
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Solving a multiobjective location routing problem with a metaheuristic based on tabu search. Application to a real case in Andalusia

Rafael Caballero [a,*], Mercedes González [a], Flor Mª Guerrero [b], Julián Molina [a], Concepción Paralera [b]

[a] Faculty of Economics, Department of Applied Economics (Mathematics), University of Malaga, Campus El Ejido, 29007 Malaga, Spain
[b] Department of Economy and Business, University Pablo de Olavide, Ctra. de Utrera, Km. 1- 41013 Sevilla, Spain

Available online 28 November 2005

Abstract

In this work we present a multiobjective location routing problem and solve it with a multiobjective metaheuristic procedure. In this type of problem, we have to locate some plants within a set of possible locations to meet the demands of a number of clients with multiple objectives. This type of model is used to solve a problem with real data in the region of Andalusia (Spain). Thus, we study the location of two incineration plants for the disposal of solid animal waste from some preestablished locations in Andalusia, and design the routes to serve the different slaughterhouses in this region. This must be done while taking into account certain economic objectives (start-up, maintenance, and transport costs) and social objectives (social rejection by towns on the truck routes, maximum risk as an equity criterion, and the negative implications for towns close to the plant).

Keywords: Location routing problems; Multiobjective combinatorial optimization; Metaheuristics

1. Introduction

In recent years, interest has increased regarding analyzing the effects of waste contamination and studying the policies required to deal with it. An example of this is the enforcement of regulations in industrialized countries to protect the natural environment and reduce ecological and personal damage derived from certain hazardous processes. Certain environmental legislation deals with transportation and waste storage, as well as with its transformation or disposal. A particular case of

* Corresponding author. Tel.: +34 95 213 1168; fax: +34 95 213 2061.
E-mail address: rafael.caballero@uma.es (R. Caballero).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2005.10.017

this in the European Union is the management of animals with Bovine Spongiform Encephalopathy (BSE).

Most livestock activity and use involves the disposal of animal remains due to death or other causes. These remains cannot be classified as urban or hazardous waste, so the producers are confronted with a service gap that must be filled.

In Spain, the tissues and organs of bovine, ovine, and caprine species are considered specific risk materials (SRM) and as such their disposal is regulated, meaning that certain procedures have to be followed.

The disposal of SRM aims at the complete destruction of risk tissues to avoid their introduction into the human and/or animal food chain, although this is an expensive process. The most feasible, economic, and viable way to do this, and remain compliant with waste legislation, is by incineration.

This work presents a model to find the best location for up to two incineration plants shared between several preestablished locations in Andalusia that will be used to dispose of solid animal waste and simultaneously find the best routes to transport the waste from each slaughterhouse to the plants opened. Thus, we are dealing with a location (deciding which plants should be opened) routing (designing the routes to transport the waste from the slaughterhouses) problem.

As we have to take into account different factors when evaluating potential locations for the new plants, we are dealing with a multiobjective problem. The literature offers various examples of the multiobjective approach being used to solve problems regarding the location of plants for the disposal of hazardous or unwanted substances (Erkut and Neuman, 1989; Giannikos, 1998; Santos et al., 2001).

Obviously, we have to include economic objectives in our study (start-up, maintenance, and transport costs), but we also take into account that the disposal of this type of waste has an associated risk. This gives rise to social rejection which can be incorporated into the model in different ways. We can find different definitions of perceived risk in the literature. Thus, Erkut and Verter (1995, 1997), and Erkut and Ingolfsson (2005) define it as individual risk multiplied by the power of the number of inhabitants in the given town, where individual risk can be expressed in relation to the probabilities of incidents and their consequences. For Giannikos (1998) perceived risk is expressed as the amount of product transported per town. We also have to take into account equity criteria in these types of problems, which refers to the equitable distribution of damage between the towns involved (Marsh and Schilling, 1993).

The problem relating to the installation of a single incineration plant has been studied in a previous work (Guerrero et al., 2005). In this work, transportation routes were not taken into account and thus the model was a discrete problem.

However, if we want more plants to be opened, try to design the routes between the plants and the slaughterhouses for the disposal of this waste, and also take into account several objectives, then, from a mathematical standpoint, we are dealing with a multiobjective location routing problem, that falls into the (more general) field of multiobjective combinatorial optimization (MOCO) problems. As can be found in Ehrgott and Gandibleux (2000), in recent years there has been a strong increase in interest in combinatorial optimization due to its potential application to real world problems. However, the optimization of multiple objectives naturally appears in most real-world applications, thus yielding MOCO problems. These types of problems combine the characteristic difficulties of combinatorial problems with the difficulties which arise from a multiplicity of objectives.

Generally, the main difficulty involved in solving this type of problem is its large computational cost. Thus, in this field, the last two decades have been highlighted by the development and improvement of approximate solution methods, usually called "heuristics and metaheuristics". These are powerful techniques generally applicable to a large number of problems, as shown in Ehrgott and Gandibleux (2000).

In our case, we are dealing, from a technical point of view (formulation and resolution) with a multiobjective location/routing (MLR) problem whose size and characteristics has not been previously studied.

In the literature, we have only found location routing (LR) problems which only include the economic objective, that is, single-objective LR problems (Albareda-Sambola et al., 2005). We have found some studies in the literature when we include multiple objectives, but these only relate to the location problem, not the routing problem (Fernández and Puerto, 2003). In practice, the location problem and the routing problem have usually been approached by solving them independently due to the computational difficulty involved, even when dealing with the economic objective only. As pointed out in Albareda-Sambola et al. (2005), solving this in two phases (a first phase to determine the plants to be opened and a second phase to determine the routes visiting each of the clients) is, in most real cases, not very useful. This is because the transport costs are determinant and this prevents us from solving the problem separately, since a bad choice of the plants in the first phase (where routes are not addressed) makes the transport costs highly non-optimal in the second phase. In short, the model must be solved in a single phase to solve this problem efficiently. For this reason, in recent years, LR problems have received greater attention in the literature, as can be appreciated in Min et al. (1998).

Generally, these problems are characterized according to the type of plants to be opened (*primary*, if they are the origin and destination of a vehicle's journeys and *secondary*, when they are only intermediate depots), according to the number of plants to be located and whether they have capacity or not, and finally, according to the number of available vehicles and whether they have capacity or not.

Exact methods for LR problems with a single primary plant have been proposed for uncapacited vehicles (Averbakh and Berman, 1994). A heuristic algorithm for the case of capacited vehicles is presented in Chien (1993). The case of multiple plants with uncapacited vehicles has also been considered in the literature. Primary locations have been located heuristically (Srivastava, 1993) and with exact methods (ReVelle et al., 1991). However, the case of multiple plants and capacited vehicles is more complex and just a few papers in the literature can be found, as can be seen in Albareda-Sambola et al. (2005), where all of them solve a single-objective LR problem. This is in fact our case, where we attempt to find solutions for multiple (uncapacited primary) plants and capacited vehicles. However, we are also dealing with a problem with multiple objectives. Thus, if we use an exact method, this will require multiple resolutions to obtain an approximation of the efficient frontier, where each resolution would offer an efficient point. On the other hand, it is a well-known fact that for this type of problem (MOCO problems), a large number of these efficient points are non-supported and obtaining them by means of an exact method requires adding constraints to the original model (see Gandibleux and Freville, 2000), therefore increasing its computational cost even more.

In short, solving this problem with an exact method is computationally too expensive or even impossible. For this reason, we opted for a metaheuristic algorithm—the MOAMP method (Caballero et al., 2004)—which is a metaheuristic algorithm for MOCO problems, based on tabu search (Glover and Laguna, 1997) which we describe in this work.

The paper is organised as follows: in the next section we describe our model, Section 3 deals with the technical aspects of the resolution procedure, Section 4 shows the application to the case of Andalusia and, finally, Section 5 presents some conclusions.

2. Description of the model

We must transport SRM waste, generated weekly by n elements belonging to the set of slaughterhouses I, to certain incineration plants located in some of the s candidate locations, where these sites form the set J. Transport passes through q towns, which are elements of the set H. This transport is carried out by means of r routes (belonging to the set R) carried out by r vehicles with a maximum capacity $CMax$ and over a time less than $TMax$. This time constraint must be included, since the working day of a lorry driver is limited.

Regarding the objectives, we take into account economic aspects as well as social ones. Thus, the first economic objective is to minimize fixed costs which include start-up and maintenance costs. This is formulated as follows:

$$\text{Min} \sum_{j=1}^{s} FC_j y_j,$$

where y_j are binary variables taking value 1 if a plant is installed in site j, $j \in J$, and 0 otherwise. These fixed costs, FC_j, include general maintenance costs, cost of the site and the land around it, building costs, machinery and transport, furniture, materials and refrigerating chambers for the correct operation of the plant, and technical services and licence costs. As the study was carried out on a weekly basis, to homogenize the various quantities that make up the costs, in our model we have financed these with a given type of interest for the period we consider the incinerator will be in service.

The second economic objective takes into account the weekly collection and transportation by lorry of the waste produced by the slaughterhouses. Thus, to the previous economic objective, fixed costs, we have to add the minimization of transport costs:

$$\text{Min} \sum_{r \in R} Cost(r),$$

where cost is measured in kilometers per route. To compute these, we assume that each of the clients is visited by exactly one route on which the lorry collects all the SRM waste from the corresponding slaughterhouse and drives it to a single plant.

In addition, the disposal of this type of waste has an associated risk that involves social rejection. This is included in our study, and comprises three different objectives.

The first one takes into account rejection by towns that trucks pass through on their way to the incineration plant:

$$\text{Min} \sum_{r \in R} Rejection(r).$$

To compute the social rejection of each route, *Rejection*(r), we took into account the towns the waste passed through and their population. This objective can be viewed as the sum, for each town, of the risk of a route passing through it, R_h, multiplied by the number of routes passing through it. This risk, R_h, will be measured as the number of inhabitants of the city.

The second social objective is an equity objective, which refers to the equitable distribution of damage between the towns involved. Thus, as a measure of equity, we minimized maximum social rejection corresponding to the town most affected by waste transportation:

$$\text{Min} \operatorname*{Max}_{h \in H} \{Risk(h)\}.$$

To compute the risk for each population, *Risk*(h), the risk occurring when a route r goes through the population h (R_h) is multiplied by the number of routes r going through this population h.

The last social objective takes into account the social rejection from towns near the incineration plant, which we have called collective disutility, and is an increasing function of town size and a decreasing function of distance from the plant to the nearby town:

$$\text{Min} \sum_{j \in J} Dis_j y_j.$$

We consider that a town is nearby if its distance from the plant is equal to or less than θ, and thus if the town is further away then the social rejection factor is not taken into account. The collective disutility caused by the plant j, Dis_j, is calculated in the following way:

$$Dis_j = \sum_{h \in H/\delta_{hj} < \theta} \frac{Pop_h}{\delta_{hj}},$$

where Pop_h is the population in town h and δ_{hj} is the distance from plant j to town h.

Our model also presents a number of additional constraints due to the real situation that it represents. First, there is a limit to the number of plants to be opened within the possible localizations, $PMax$, and we have to bear in mind the maximum capacity of each truck, $CMax$, and the maximum number of hours that a route can take, $TMax$.

3. MOAMP design for the MLR problem

For the resolution of this problem a metaheuristic algorithm, MOAMP (Caballero et al., 2004), has been used. This is a metaheuristic for the resolution of MOCO problems based on tabu search (TS). MOAMP (multiobjective metaheuristic using an adaptative memory procedure) tries to adapt a tabu search procedure to the structure of the efficient set of a multiobjective problem. In this sense, it is a well-known fact that the efficient points of a MOCO problem are "connected", that is, any efficient point is close enough to another efficient point; by "close enough" we mean that in a not too extensive neighborhood of this efficient point another efficient point can be found. This proximate optimality principle of the efficient points of a multiobjective problem will be the main point in the MOAMP method. Thus, MOAMP generates, by means of some tabu searches, an initial set of efficient points (Phase I of the algorithm) and by use of these tries to obtain a good approximation of the rest by means of an intensification process (Phase II of the algorithm).

To build this initial set of efficient points it carries out a series of linked tabu searches (linked means that the initial point of each one will be the last point visited by the previous search) where each point visited on each iteration could be a part of the final approximation obtained. That is, additional efficient solutions may be found during this phase because all visited points are checked for inclusion in the list of efficient points (LE). By means of this continuous updating process of the efficient points list, we obtain a list of all the possible efficient points visited by means of each one of the tabu searches.

The functions to optimize regarding these TS are the following:

- $p+1$ tabu searches where, in each case, the function to optimize is the ith objective function and the first function will be optimized again in the last place, to complete a cycle. By means of these $p+1$ searches we try to find a set of efficient points approaching the p optima of each one of the objectives, as well as the possible efficient points on the way from an optimal point to the following one.
- Randomly generate N weighting vectors $\lambda = (\lambda_1, \ldots, \lambda_p)$ and use these to make N tabu searches where the function to minimize is the following:

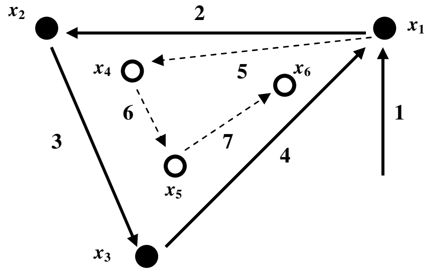
$$F_\lambda(x) = \max\left\{\lambda_i\left(\frac{f_i^{\max} - f_i(x)}{f_i^{\max} - f_i^{\min}}\right); \quad i = 1, \ldots, p\right\},$$

where

- $f_i^{\max}$ is the maximum value of the ith objective over the efficient points set obtained up to now.
- $f_i^{\min}$ is the minimum value of the ith objective over the efficient points set obtained up to now.
- N then represents maximum number of tabu searches that could be carried out without any change in the LE.

By means of these last N searches MOAMP is trying to find the point x minimizing the L_∞ distance (normalized by the range of each function inside the previously obtained approximation of the efficient set and by the weighting vector λ) to the point $(f_1^{\max}, \ldots, f_p^{\max})$. This point, containing the best individual (assuming a maximization problem) values for each objective, is known as the ideal point and it is also a well-known fact that if a point minimizes this normalized L_∞ distance to the ideal point, then it is an efficient point. In general, a point minimizing an L_q normalized distance, $q \in [1, \infty]$, to the ideal point, is also an efficient point, and the set of all the efficient points obtained this way is known as the compromise set. These points have the common characteristic of representing a good balance between the objectives; this is, they are points that, without being very good regarding certain objectives, offer good value regarding them all as a whole. The aim of looking for efficient points in the compromise set is try to obtain a sufficiently diverse sample of efficient points such that, when intensifying the search on these points, the widest possible approximation of the efficient set is obtained.

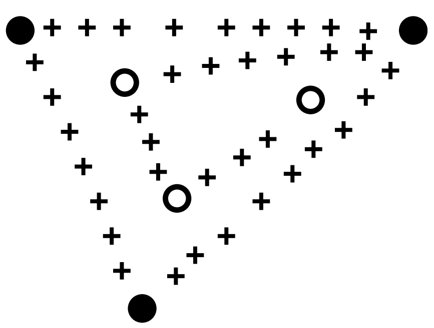
This set of tabu searches allows us to complete the widest possible initial set of efficient points, where these different searches explore different areas of the efficient set. Thus, by linking the different tabu searches, we seek to carry out an exploration of the efficient set as shown in Graph 1, for a case with three objectives. In this example, MOAMP carries out an initial TS (arch no. 1) leading us to the optima of the first function (point



Graph 1. Linked TS.

x_1), from which we move to the optima of the other two functions (arch no. 2, point x_2, arch no. 3, point x_3) and return using arch no. 4 to the optima of the first function. Finally, from this point x_1, we link three more searches (arches 5, 6, and 7) to visit the compromise points x_4, x_5 and x_6. During these searches, as pointed out, efficiency is verified for each visited point and the list of efficient points is constantly updated. Consequently, at the end of these $p + 1 + N$ tabu searches we obtain a sample of the efficient set distributed by the areas where one of the objectives is predominant, as well as for those areas characterized by a balance among the different objectives. In our example with three objectives, this sample could be as shown in Graph 2.

Once this set of efficient points is obtained, the search is intensified on it. By means of this last stage of the algorithm, MOAMP carries out an intensification around all the efficient points of the initial sample, generating a new set of efficient points, where this process is repeated until this intensification no longer offers any new efficient points. Thus, it "fills the gaps" in the efficient set between the elements of the initial set.



Graph 2. Visited points are sent to LE.

Phase I and II of MOAMP form a *cycle* in the execution, but MOAMP presents the possibility of executing several *cycles* (launch Phase I and II again) where LE is used as a long-term memory and then searches concentrate on the non-visited areas.

To check the efficiency of MOAMP the authors solved three different sets of test problems studied in the literature (related to three different types of MOCO problems). These sets included bi-objective multidimensional knapsack problems (biMKP), bi-objective assignment problems (biAP), and bi-objective set packing problems (biSPP), respectively, obtaining promising computational results as well as high-quality approximations of the efficient sets for the three types of problems. The input data used can be found on the MCDM Numerical Instances Library web pages: http://www.terry.uga.edu/mcdm/index.html.

The quality of these approximations were measured according to three different measures widely used in the MOCO literature:

- M1 (Ulungu, 1993): the proportion of exact efficient solutions found.
- M2 (Ulungu, 1993): the proportion of solutions in PE (the approximation found) which are at least located in the triangles generated in the objective space by two successive supported efficient solutions.
- SSC (Zitzler and Thiele, 1999): the size of the space covered. SSC measures the proportion of the volume of the dominated points.

These results are summarized in Tables 1–3.

Then, this results show that MOAMP is not highly depending on the special structure of the problem and can solve efficiently three completely different kinds of problems within a good computational times. And this is one of the main points of MOAMP: only the codification of solutions and neighborhood definition were changed when solving the three different types of problems. This was the main reason to choose MOAMP to solve our problem, because, as mentioned, we only had to change the codification of solutions and neighborhood definition to use this method with our model.

Table 1
Results obtained with MOAMP for the biAP

Instances	# Exact	# MOAMP	M1(%)	M2 (%)	SSC (%)
2AP5-1A20	8	8	100	100	100
2AP10-1A20	16	16	100	100	100
2AP15-1A20	39	39	97.44	100	99.9
2AP20-1A20	55	50	80	100	99.8
2AP25-1A20	74	75	58.11	90.67	99.6
2AP30-1A20	88	79	46.07	87.34	99.5
2AP35-1A20	81	76	28.4	73.68	99.3
2AP40-1A20	127	111	13.28	64.86	98.9
2AP45-1A20	114	109	0	55.96	98.8
2AP50-1A20	163	156	0	36.54	98.5
2AP60-1A20	128	114	0	11.4	98.1
2AP70-1A20	174	150	0.57	20	98
2AP80-1A20	195	158	0	4.43	97.9
2AP90-1A20	191	173	0	1.16	97.3
2AP100-1A20	223	223	0	0	97.8

Table 2
Results obtained with MOAMP for the biMKP

Instances	# Exact	# MOAMP	M1 (%)	M2 (%)	SSC (%)
2MKP50-1A	34	33	94.12	100	99.9
2MKP100-1A	171	154	54.39	89.61	99.9
2MKP150-1A	248	180	36.69	90.56	99.9
2MKP200-1A	444	303	5.41	70.96	99.9
2MKP250-1A	643	444	10.73	74.77	99.9
2MKP300-1A	754	499	4.77	76.35	99.8
2MKP350-1A	966	536	0.72	60.07	99.9
2MKP400-1A	1175	604	2.13	66.23	99.9
2MKP450-1A	1830	636	0	51.57	99.8
2MKP500-1A	1922	674	0	47.03	99.8
2MKP100	121	104	56.2	93.27	99.9
2MKP250	567	308	0.53	51.3	99.9
2MKP500	1427	551	0	26.13	99.6

On the other hand, the MOAMP parameters are few and simple. In the basic design, the user must determine the number of iterations of the TSs (*iter*), the maximum number of tabu searches that could be carried out without any change in LE (*Max_searches*), and the number of cycles to be carried out (*cycles*).

In this way, for our particular situation—an MLR problem with some additional constraints—a representation of the solutions and a set of neighbourhoods was designed. Regarding the representation, a solution x_0 is represented by a list of trucks:

$\{T_s\}, s = 1, \ldots, NumTrucks_i$.

Each one has an ordered list of clients, OLC_s, representing the clients (and the order to be visited) that it must visit and a plant P_s where the route begins and finishes. Regarding this representation, we consider the following set of movements from a given point x_0, ordered into two hierarchies: movements relating to the routes ($Mov1$, $Mov2$, and $Mov3$) and movements relating to the locations of the plants ($Mov4$, $Mov5$, and $Mov6$), where the latter three have been developed according to the ideas of Albareda-Sambola et al. (2005).

Table 3
Results obtained with MOAMP for the biSPP

Instances	# Exact	# MOAMP	M1 (%)	M2 (%)	SSC (%)
2MIS100-300A	41	40	90.24	100	99.9
2MIS101-300A	22	22	100	100	100
2MIS100-500A	17	17	94.12	94.12	99.9
2MIS101-500A	21	21	95.24	100	99.9
2MIS200-600A	77	75	45.45	62.67	97.5
2MIS201-600A	96	78	56.25	85.9	97.9
2MIS200-1000A	43	36	65.12	100	99.5
2MIS201-1000A	45	16	20	75	97.3
2SPP100-300A	10	8	30	37.5	99.1
2SPP101-300A	9	9	77.78	77.78	99
2SPP100-500A	14	8	42.86	100	99.8
2SPP101-500A	12	9	66.67	88.89	94.6
2SPP200-600A	38	33	55.26	69.7	96.2
2SPP201-600A	54	27	20.37	62.96	99.5
2SPP202-600A	13	12	92.31	100	99.4
2SPP203-600A	14	9	7.14	44.44	94.8
2SPP200-1000A	25	11	40	90.91	98.7
2SPP202-1000A	8	5	62.5	100	99.1
2SPP203-1000A	16	9	25	55.56	92.1

- *Movements relating to the routes.*
- *Mov*1: A client i of a truck T_s is transferred to a different truck $T_{s'}$.
- *Mov*2: A client i of a truck T_s is extracted and a new truck $T_{s'}$ is created for this client.
- *Mov*3: A truck T_s is eliminated and all its clients are re-allocated among the remaining trucks.
- *Movements relating to the locations of the plants.*
- *Mov*4: One of the opened plants P_j is closed, and all the clients that were visited by routes with their origin and destination in the eliminated plant are re-allocated among the remaining routes.
- *Mov*5: One of the opened plants P_j is closed and a closed plant $P_{j'}$ is opened, and an attempt is made to assign all the routes corresponding to the eliminated plant to the new opened plant, $P_{j'}$. When a route cannot be assigned to the new plant (because it would violate the time limit for routes), the clients of this route are re-allocated among the remaining routes.
- *Mov*6: One of the closed plants P_j is opened and all the routes are re-created taking into account the re-opened plant.

Regarding the movements relating to the routes (*Mov*1, *Mov*2, and *Mov*3), once the set of clients is associated to a given truck T_s as well as a plant from which the truck originates, the order of the clients in the route is determined using a widely used and validated method. This is the heuristic NN (nearest neighbour, Flood, 1956) for the TSP that offers a computing time of $O(n^2)$, n being the number of clients on the route, and a bound for the gap to the optimal solution (worst case guarantee) of $0.5(\log_2(n) + 1)$.

For the resolution of the problem, the algorithm has been reinforced by computing good initial points (seeds) by means of a greedy function to improve the performance of the tabu searches which looks for the optimal solution of each objective. These seeds are built in the following way:

- All possible combinations of two opened plants are selected and their routes are generated with a greedy function.
- For each objective we choose as its seed the combination offering the smallest value for this objective.

  The key to this process is the greedy function that is able to compute good routes very quickly once two opened plants are selected. The way this greedy function builds the routes is as follows:

- Repeat while the number of clients not assigned to a route is greater than zero:
  - Select a client at random Cl_i without being assigned to a route.
  - Create a new truck T_s for this client, departing from the plant nearest to this client.
  - Make $TruckIsFull = 0$, $LastIn = Cl_i$.
  - Repeat while $TruckIsFull = 0$.
  - Select the client $Cl_{i'}$ (without being assigned to a route) closest to $LastIn$.
  - Try to include the client $Cl_{i'}$ in the truck T_s.
    - If it is possible to include the client $Cl_{i'}$ in the truck T_s, then $TruckIsFull = 0$, $LastIn = Cl_{i'}$.
    - If it is not possible to include the client $Cl_{i'}$ in the truck T_s, then $TruckIsFull = 1$.

This seeding procedure improved the performance of the TS associated with the p objectives of the problem, regarding both computational time and best value found. However, as we only need one initial point for the first tabu search (for the rest of the searches the initial point is the last point visited by the previous search) we only have to decide which objective should be the first one and use its seed as the initial point. In our case, the first objective is the transport cost, because it is the most difficult objective to optimize and its seed quality is so high.

4. Application to the case in Andalusia

At the time of our study, there were 93 slaughterhouses in Andalusia generating SRM waste, which has to be completely incinerated according to the law. The following towns were chosen as candidates for plant location: (1) Antequera, as this town is the geographical centre of Andalusia; (2) Aznalcollar and Alquife, since they have a large number of unemployed people—mainly lorry drivers and machine engineers—due to relatively recent mine closures; (3) Olvera and Alcala la Real, because of their proximity to several Andalusian provinces; and (4) Osuna, since there is a residual transformation plant in this town where any type of SRM residuals can be transformed. Thus, we would like to analyze whether this location could be used as an incineration plant.

In short, we face a problem with 6 possible locations (where at most we can choose two of them), 93 clients to serve, 5 objectives to optimize, and constraints on truck capacity and on the duration of the route.

Regarding the remaining parameters of the model:

- For the collective disutility, we fixed the θ parameter at 10 km.
- The maximum number of plants to be opened, $PMax$, is 2.
- The maximum capacity of each truck, $CMax$, is 3500 kg.
- The maximum number of hours that a route can take, $TMax$, is 8. To compute the time of each route, we allow 1 hour to load the truck, $TLoad$, for each client on the route.

This algorithm has been implemented using the C++ 6.0 language and the Visual Microsoft C++ compiler, on a PC with a PentiumIV 2.4 GHz processor for all the runs. Several runs were made to adjust parameters and compare several approximations. We placed no constraints on the computational time needed to solve the problem, because it was going to be solved only once to decide which plants should be opened. Thus, our only criteria to adjust the parameters of MOAMP was to obtain the best quality in the approximation, where the time needed was of no importance. Therefore, each time a run was done we compared the efficient points obtained to check whether any of them was dominated by a point of a previous run or whether the best value found for each objective (in every run) was not achieved, until we obtained an approximation without a dominated point and without a decrease in the best value found for the five different objectives. Finally, the MOAMP parameters were the following: the number of iterations of the TSs $iter = 500$, the maximum number of tabu searches that could be carried out without any change in LE $Max\_searches = 3$, and the number of cycles to be carried out $cycles = 1$. For the final run, the run selected, the algorithm took 413 seconds to find 335 efficient points. As there are no instances of this problem available and there is no other method to compare with in the literature, we tried to compare this performance with the results obtained in test problems in Albareda-Sambola et al. (2005), in spite of the problems solved are quite different as is shown in Table 4.

First of all, we noted that most of these test problems could not be solved within 48 hour with CPLEX, and this alerted us to the fact that an exact method could not be expected to be able to solve our problem. If we consider the computational times presented in Albareda-Sambola et al. (2005), where they report the time to obtain the best solution found (smaller than the total time), the problem is single-objective and the routes are not limited by time, for a set of instances with 10 plants and 20 and 30 clients, the average computational time for obtaining the best solution found is between 15.52 and 52.26 seconds (computed on a SUN sparc station 10/30), depending on the problem. We needed 413 seconds to obtain 335 points (this is, 1.23 seconds per solution), and this leads us to the conclusion that our algorithm is computationally efficient. Then, we are comparing with a similar problem in the literature: it is a single-objective problem, the size of instances is smaller and we have some more constraint, but our computational performance looks good if we take into account the ratio (time consumed/number of solutions obtained) and compare it with the time needed to get a solution in Albareda-Sambola et al. (2005).

However, more experiments were carried out to check the efficiency of the proposed algorithm. This way, we adapted our algorithm in order to solve some of the problems used in Albareda-Sambola et al. (2005). Then, firstly we solved the problems as they were, this is, with a single criterion, and compared computational times (where we compare our total time with their time to obtain the best solution found) and quality of the solution. In this last case, quality of the solution is measured as the percentage of overall the capacity of the set of open plants that is consumed by the clients' demands. Table 5 shows a summary of these results, where columns headed by A-S include the results in Albareda-Sambola et al. (2005) and columns headed by LR-M include the results of the location routing feature of MOAMP method.

These results highlight the LR-M performance, showing that this method is able to maintain the same quality as A-S method but within less computational time. Moreover, we added some (conflicting) objectives to these instances, in order to get problems with 2 and 4 objectives. Our aim now is checking how the performance of LR-M behaviors as the number of objectives is increased. Then, we show the total time to solve the multiobjective instances as well as the ratio time/#solutions. These results are shown in Table 6.

As it is shown, LR-M maintains the ratio time/#solutions within reasonable bounds and even it remains close to the ratio obtained for the application solved (1.23 seconds per solution), in spite of the differences between these problems. Then, we

Table 4
Comparison with test problems

	Application to Andalusia	Test problems
Locations	6	5–10
Clients	93	10–30
Objectives	5	1
Constraints	Max time for routes capacited vehicles	Capacited vehicles

Table 5
Comparison for the single-objective instances

Problem	A-S time (seconds) to best sol.	LR-M time (seconds) total	A-S % used cap	LR-M % used cap
S1a	0.28	0.27	85.49	86.03
S1b	0.27	0.23	88.45	90.05
S1c	2.00	0.22	95.79	89.70
S2a	1.20	0.86	84.59	91.33
S2b	1.20	0.66	80.32	91.61
S2c	8.11	0.61	93.19	91.91
S3a	7.93	2.15	83.47	80.49
S3b	7.06	1.50	72.85	95.22
S3c	18.23	1.29	87.27	89.24
M1a	15.52	1.29	96.57	98.75
M1b	14.20	1.15	98.13	95.72
M1c	13.51	1.13	97.82	92.06
M2a	38.80	2.58	95.85	97.88
M2b	52.69	2.08	95.87	96.34
M3c	41.98	1.94	99.66	96.86

Table 6
Comparison for the multiobjective instances

Problem	1-obj time (total)	2-obj time (total)	time/ #sol.	4-obj time (total)	time/ #sol.
S1a	0.27	0.57	0.14	0.82	0.16
S1b	0.23	0.48	0.10	0.70	0.14
S1c	0.22	0.46	0.15	0.68	0.23
S2a	0.86	3.07	1.02	5.32	1.33
S2b	0.66	2.35	0.47	4.00	0.80
S2c	0.61	2.24	0.75	3.77	0.75
S3a	2.15	10.20	3.40	18.39	4.60
S3b	1.50	6.62	1.66	13.95	2.79
S3c	1.29	6.61	2.20	11.70	2.92
M1a	1.29	4.00	0.57	6.26	0.52
M1b	1.15	3.50	0.50	5.53	0.79
M1c	1.13	8.20	1.03	5.19	0.58
M2a	2.58	10.53	1.50	18.07	2.26
M2b	2.08	8.86	1.48	15.36	1.92
M3c	1.94	8.21	1.17	14.07	2.01

conclude our procedure LR-M is computationally efficient and stable.

Finally, a summary of the results obtained for the application is shown in Table 7. This presents examples of efficient solutions with their normalized values for each objective in the first five columns, where the value 0 represents the minimum for this objective, that is, the minimum value found, whereas the value 1 is the biggest value found. Thus, solution 1 minimizes the fixed costs, solution 4 minimizes transport costs, solution 80 minimizes social rejection, maximum risk is minimized with solution 90, and collective disutility is minimized with solution 65.

The next two columns include the sum of the normalized values for each objective and the maximum of these values, respectively. Thus, the last two solutions in the table represent the minimum among all the 335 efficient solutions for each of these two concepts: solution 281 minimizes the sum of the normalized values and solution 35 achieves the minimum of the maximum normalized values.

The following six columns show the six candidate locations along with the number of slaughterhouses where collections are done and the number of plants opened for each solution. Thus, solution 1 opens only one plant in Alquife, and another is also opened in solution 35, but in Aznalcollar. On the other hand, some other efficient solutions

Table 7
Several solutions

Sol.	Fixed costs	Transport cost	Social rejection	Max risk	Coll. disutility	Aggregated	Max	Aznal.	Antequera	Alcala real	Olvera	Alquife	Osuna	#Plants	# Trucks
Sol1	0	0.789	0.181	0.247	0	1.217	0.789	0	0	0	0	93	0	1	59
Sol4	0.949	0	0.162	0.353	0.464	1.929	0.949	38	0	55	0	0	0	2	59
Sol80	0.993	0.206	0	0.247	0.797	2.244	0.993	18	75	0	0	0	0	2	60
Sol90	0.852	0.475	0.214	0	0.304	1.848	0.852	0	0	0	0	58	35	2	59
Sol65	0	0.789	0.181	0.247	0	1.217	0.789	0	0	0	0	93	0	1	59
Sol281	0	0.786	0.182	0.247	0	1.215	0.786	0	0	0	0	93	0	1	59
Sol35	0.035	0.422	0.425	0.353	0.085	1.323	0.425	93	0	0	0	0	0	1	58

open two plants, e.g., solution 4, where 38 slaughterhouses have collections from Aznalcollar and 55 from Alcala la Real. Finally, the last column includes the number of trucks needed for the routes in the solution.

In summary, in this table we present the decision-maker with a set of solutions representing different situations. The first five solutions represent the optimal solutions for the five different criteria whereas the last two represent well-balanced ones, that is, they do not yield really good performance for a given objective but neither are they inadequate. The decision-maker can consider other relevant aspects regarding the solutions in this table, such as the number of plants opened, the number of clients collected by each plant, and the number of trucks needed to serve the clients. The decision-maker can also obtain information about the trade-off between the objectives for each solution, because we show him/her the normalized values instead of the real values. Thus, he/she can see that solution 4, for example, is optimal for transport costs, good regarding social rejection, not bad for the last two objectives, but really bad regarding fixed costs.

Finally, we want to point out that the decision-maker is not obliged to choose any of these initial seven solutions. If he/she is not comfortable with any of these solutions we can present a different set which includes some of the other 328 remaining solutions found.

Once the decision-maker chooses a solution, we offer him/her complete information about the routes, the clients on each route, the duration of the route, the number of kilos that any truck is going to load, etc. Thus, we offer all the information needed to implement the routes.

5. Conclusions

In this work we have studied a problem that has not been previously dealt with in the literature, i.e., a multiobjective location routing with a capacited vehicle and a constraint on the time that a route can take. To solve this problem we adapted a metaheuristic for MOCO problems based on tabu search, the MOAMP method. This adaptation consists in some specific neighborhood definitions inspired by the movements used for a similar problem in the literature, a single-objective location routing problem with capacited vehicles. The efficiency of our approach has been shown compared to the results obtained for a similar problem in the literature.

This metaheuristic procedure has been applied to a real problem, the location of two incineration plants and the transport of animal waste in Andalusia, obtaining a representative and broad approximation of the efficient set of this problem.

Future work includes the attempt to apply this model to the remaining regions in Spain. To this end, some new technical aspects will be included in the metaheuristic procedure, such as fuzzy information as well as an interactive mode to let the real decision maker play a more rich rule in the process.

Acknowledgements

This research has been partially founded by the research projects of Andalusian Regional Government and Spanish Ministry of Educacion y Ciencia.

References

Albareda-Sambola, M., Díaz, J., Fernández, E., 2005. A compact model and tight bounds for a combined location/routing problem. Computers and Operations Research 32 (3), 407–428.

Averbakh, I., Berman, O., 1994. Routing and location-routing p-delivery men problems on a path. Transportation Science 28, 162–166.

Caballero, R., Gandibleux, X., Molina, J. 2004. MOAMP—A multiobjective metaheuristic using an adaptative memory procedure. Technical Report. University of Valenciennes.

Chien, T., 1993. Heuristic procedures for practical sized uncapacited location–capacited vehicles routing problems. Decision Sciences 24, 995–1021.

Ehrgott, M., Gandibleux, X., 2000. A survey and annotated bibliography on multiobjective combinatorial optimization. OR Spektrum 22, 425–460.

Erkut, E., Ingolfsson, A., 2005. Transport risk models for hazardous materials: Revisited. Operations Research Letters 33, 81–89.

Erkut, E., Neuman, S., 1989. Analytical models for locating undesirable facilities. European Journal of Operational Research 40, 275–291.

Erkut, E., Verter, V., 1995. Hazardous materials logistics. In: Facility Location. A Survey of Applications and Methods. Springer, Heidelberg.

Erkut, E., Verter, V., 1997. Modeling of transport risk for hazardous materials. Operations Research 46 (5), 625–642.

Fernández, E., Puerto, J., 2003. Multiobjective solution of the uncapacitated plant location problem. European Journal of Operational Research 145, 509–529.

Flood, M., 1956. The travelling salesman problem. Operations Research 4, 61–75.

Gandibleux, X., Freville, A., 2000. Tabu search based procedure for solving the 0–1 multiobjective knapsack problem: The two objectives case. Journal of Heuristics 6, 361–383.

Giannikos, I., 1998. A multiobjective programming model for locating treatment sites and routing hazardous wastes. European Journal of Operational Research 104, 333–342.

Glover, F., Laguna, M., 1997. Tabu Search. Kluwer Academic Publishers, Boston.

Guerrero, F., Paralera, C., Caballero, R., González, M., Molina, J., 2005. Location of specific risk material incineration plants in Andalusia using a multicriteria approach. Investigación Operacional 26 (2), 135–141.

Marsh, M., Schilling, D.A., 1993. Equity measurement in facility location analysis: A review and framework. European Journal of Operational Research 74, 1–17.

Min, H., Jayaraman, V., Srivastava, R., 1998. Combined location routing problems: A synthesis and future research directions. European Journal of Operational Research 108, 1–15.

ReVelle, C., Cohon, J., Shobrys, D., 1991. Simultaneous siting and routing in the disposal of hazardous wastes. Transportation Science 25, 138–145.

Santos, D.R., Suárez-Vega, R., Dorta, P., 2001. Un Modelo de Decisión Multicriterio para Localización de Centros de Tratamientos de Residuos. Estudios de Economía Aplicada 17, 163–182.

Srivastava, R., 1993. Alternate solution procedures for the location routing problem. Omega 21, 497–506.

Ulungu, E. 1993. "Optimisation combinatoire multicritere: D'etermination de l'ensemble des solutions efficaces et methodes interactives, Ph.D. Thesis, Universite de Mons-Hainaut, Faculte des Sciences, p. 313.

Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation 3 (4), 257–271.