J Math Model Algor (2008) 7:59–78
DOI 10.1007/s10852-007-9073-6

A Particle Swarm Optimization Algorithm with Path Relinking for the Location Routing Problem

Yannis Marinakis · Magdalene Marinaki

Received: 5 June 2007 / Accepted: 28 September 2007 /
Published online: 16 November 2007

Abstract This paper introduces a new hybrid algorithmic nature inspired approach based on particle swarm optimization, for solving successfully one of the most popular logistics management problems, the location routing problem (LRP). The proposed algorithm for the solution of the location routing problem, the hybrid particle swarm optimization (HybPSO-LRP), combines a particle swarm optimization (PSO) algorithm, the multiple phase neighborhood search – greedy randomized adaptive search procedure (MPNS-GRASP) algorithm, the expanding neighborhood search (ENS) strategy and a path relinking (PR) strategy. The algorithm is tested on a set of benchmark instances. The results of the algorithm are very satisfactory for these instances and for six of them a new best solution has been found.

Keywords Particle swarm optimization · MPNS-GRASP · Path relinking · Expanding neighborhood search · Location routing problem

Mathematics Subject Classifications (2000) 90B06 · 90B80 · 90C59 · 90C27

1 Introduction

Several biological and natural processes have been influencing the methodologies in science and technology in an increasing manner in the past years. Feedback control processes, artificial neurons, the DNA molecule description and similar

Y. Marinakis (✉) · M. Marinaki
Department of Production Engineering and Management,
Technical University of Crete, 73100 Chania, Greece
e-mail: marinakis@ergasya.tuc.gr

M. Marinaki
e-mail: magda@dssl.tuc.gr

genomics matters, studies of the behaviour of natural immunological systems, and more, represent some of the very successful domains of this kind in a variety of real world applications. During the last decade, nature inspired intelligence became increasingly popular through the development and utilisation of intelligent paradigms in advanced information systems design. Cross-disciplinary team-based thinking attempts to cross-fertilise engineering and life science understanding into advanced inter-operable systems. The methods contribute to technological advances driven by concepts from nature/biology including advances in structural genomics (intelligent drug design through imprecise data bases), mapping of genes to proteins and proteins to genes (one-to-many and many-to-one characteristics of naturally-occurring organisms), modelling of complete cell structures (showing modularity and hierarchy), functional genomics (handling of hybrid sources and heterogeneous and inconsistent origins of disparate databases), self-organization of natural systems, etc. Among the most popular nature inspired approaches, when the task is optimization within complex domains of data or information, are those methods representing successful animal and micro-organism team behaviour, such as swarm or flocking intelligence, artificial immune systems, optimized performance of bees, or ant colonies, etc.

In this paper, we demonstrate how a nature inspired intelligent technique, the particle swarm optimization (PSO) [23] and three metaheuristic techniques, the multiple phase neighborhood search – greedy randomized adaptive search procedure (MPNS-GRASP) [38], the expanding neighborhood search (ENS) [36] and the path relinking [18] can be incorporated in a hybrid scheme, in order to give very good solutions in the location routing problem (LRP). Particle swarm optimization algorithms, due to their simplicity in coding and their global and local exploration abilities, are usually applied, with remarkable results, in continuous optimization problems. In this paper, we focus in the way a PSO algorithm can easily and efficiently be applied in a classic combinatorial optimization problem, like the location routing problem. The main advantage of the application of a PSO algorithm in the LRP is that, contrary to other metaheuristics, there are only two variables for each member of the population that will have to be calculated in each iteration, the position and the velocity. Thus, the combination of a PSO algorithm with a very fast local search strategy, like the expanding neighborhood search strategy [36, 37], will lead to very fast and efficient algorithm and will reduce, significantly, the computational time of the algorithm making the algorithm suitable for solving very large scale location routing, and more difficult combinatorial optimization, problems in short computational time. The rest of the paper is organized as follows: In the next section, a description of the location routing problem is presented. In the third section, the proposed algorithm, the hybrid particle swarm optimization (HybPSO-LRP), is presented and analyzed in detail. An analysis of the computational results is presented in the fourth section while in the last section conclusions and future research are given.

2 The Location Routing Problem

In the last few years, the need of an integrated logistic system has become a primary objective of every manager in a company. Managers recognize that there is a strong relation between the location of facilities, the allocation of suppliers, vehicles and

customers to the facilities and in the design of routes around the facilities. In a location routing problem the optimal number, the capacity and the location of facilities are determined and, also, the optimal set of vehicle routes from each facility is sought.

In most location models, it is assumed that the customers are served directly from the facilities being located. Each customer is served on his or her own route. In many cases, however, customers are not served individually from the facilities. Rather, customers are consolidated into routes which may contain many customers. In the location routing problem, a number of facilities are located among candidate sites and delivery routes are established to a set of users in such a way that the total system cost is minimized. As Perl and Daskin [47] pointed out, location routing problems involve three inter-related, fundamental decisions: where to locate the facilities, how to allocate customers to facilities, and how to route the vehicles to serve customers.

The difference of the location routing problem from the classic vehicle routing problem is that not only routing must be designed but also, the optimal depot location must be simultaneously determined. The main difference between the location routing problem and the classical location – allocation problem is that, once the facility is located, the former requires a visitation of customers through tours while the latter assumes that the customer will be visited from the vehicle directly and, then, it will return to the facility without serving any other customer [42]. In general terms, the combined location routing model solves the joint problem of determining the optimal number, capacity and location of facilities serving more than one customer and finding the optimal set of vehicle routes. In the location routing problem, the distribution cost is decreased due to the assignment of the customers to vehicles while the main objective is the design of the appropriate routes of the vehicles.

The location routing problem can be stated as follows: Let $G = (N, E)$ be an undirected graph, where $N = \{1, ..., n\}$ is the set of nodes and E is the set of edges. Each node can be used either as facility node or customer node or both. Let $C = (c_{fj})$ be a matrix of costs, distances or travel times associated with the number of edges. If $c_{fj} = c_{jf}$ for all $f, j \in N$, the matrix and the problem is said to be symmetrical, otherwise it is asymmetrical. C satisfies the triangle inequality if and only if $c_{fh} + c_{hj} \geq c_{fj}$ for all $f, j, h \in N$. There can be, at most, k identical vehicles of capacity Q_k based at facility j. It is assumed here that c_{fj} are nonnegative and that all vehicles have the same capacity Q. Every customer has a nonnegative demand q_f that must be served by one single vehicle. Each facility has an opening cost F_j. The number of vehicles used is unknown and is a decision variable. Each route must begin and end at the same depot and its total load must not exceed vehicle capacity and the total load of the routes assigned to a depot must fit the capacity of that depot. The total cost of a route includes the costs of traversed edges. The objective is to find which depots should be opened and which routes should be constructed, in order to minimize the total cost (fixed costs of depots, plus total cost of the routes).

The location routing problem is very difficult to solve with the use of exact algorithms, especially if the number of customers or the candidate for location facilities are very large due to the fact that this problem belongs to the category of *NP-hard problems*, i.e. no polynomial time algorithms are known for their solution. However, a number of exact algorithms have been proposed for the solution of this problem solving mainly problems with a small number of locations and customers [2, 17, 26–29, 41, 55, 60].

As it is an NP-hard problem, the instances with a large number of customers cannot be solved to optimality within reasonable time. Thus, many heuristic, metaheuristic and stochastic algorithms have been developed in order to find a near optimal solution in reasonable computational time. An analytical survey of the location routing algorithms can be found in [34, 42, 44]. Classic heuristic algorithms for the solution of the problem can be found in [1, 3, 4, 6, 7, 10, 12, 22, 25, 33, 46, 52, 58]. Several metaheuristic algorithms have been proposed for the solution of the location routing problem. In [5, 8, 32, 40, 51, 57] algorithms based on Tabu Search are presented. Simulated annealing for location routing is used in [31, 32, 59] while in [48] a location routing algorithm based on greedy randomized adaptive search procedure (GRASP) is applied. Genetic algorithms are used in [21, 49]. Variable neighborhood search algorithms for the LRP are presented in [40].

3 Hybrid Particle Swarm Optimization for the Location Routing Problem

3.1 Fundamentals of Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based swarm intelligence algorithm. It was originally proposed by Kennedy and Eberhart as a simulation of the social behavior of social organisms such as bird flocking and fish schooling [23]. PSO uses the physical movements of the individuals in the swarm and has a flexible and well-balanced mechanism to enhance and adapt to the global and local exploration abilities. Because of its easy implementation and inexpensive computation, its simplicity in coding and consistency in performance, PSO has proved to be an effective and competitive algorithm for optimization problems in continuous space. Most applications of PSO have concentrated on the optimization in continuous space while recently, some work has been done on discrete optimization problems.

Since its introduction, PSO has gained rapid popularity and has proved to be a competitive and effective optimization algorithm in comparison with other metaheuristics. The PSO algorithm first randomly initializes a swarm of particles. The position of each individual (called a particle) is represented by a d-dimensional vector in problem space $s_i = (s_{i1}, s_{i2}, ..., s_{id})$, $i = 1, 2, ..., B$ (B is the population size), and its performance is evaluated on the predefined fitness function. Thus, each particle is randomly placed in the d-dimensional space as a candidate solution. One of the key issues in designing a successful PSO for the location routing problem is to find a suitable mapping between location routing problem solutions and particles in PSO. Each particle i is recorded via the path representation of the tour, that is, via the specific sequence of the nodes and an index indicating to which facility this tour is assigned. Thus, for each particle a one dimensional array of the form $i = (h_1, h_2, ..., h_{l-1}, z_1, h_l, h_{l+1}, ..., z_2, ...h_n, z_m)$ is created where $h_1, h_2, ..., h_{l-1}$ are the nodes that correspond to the tours of the location indicated by the index z_1, $h_l, h_{l+1}, ...$, are the nodes that correspond to the tours of the location indicated by the index z_2, and so on, n is the number of nodes and m is a variable which corresponds to the number of open facilities.

The velocity of the i-th particle $v_i = (v_{i1}, v_{i2}, ..., v_{id})$ is defined as the change of its position. The flying direction of each particle is the dynamical interaction of individual and social flying experience. The algorithm completes the optimization

through following the personal best solution of each particle and the global best value of the whole swarm. Each particle adjusts its trajectory toward its own previous best position and the previous best position attained by any particle of the swarm, namely p_i and p_g. In each iteration, the swarm is updated by the following equations [23]:

$$v_i(t+1) = v_i(t) + c_1 r \text{ and } 1(p_i - s_i(t)) + c_2 r \text{ and } 2(p_g - s_i(t)) \tag{1}$$

$$s_i(t+1) = s_i(t) + v_i(t+1) \tag{2}$$

where, $p_i = (p_{i1}, ..., p_{id})$ is the best position encountered by i-th particle so far; p_g represents the best position found by any member in the whole swarm population; t is iteration counter; c_1 and c_2 are acceleration coefficients; *rand*1 and *rand*2 are two random numbers in [0, 1]. Acceleration coefficients c_1 and c_2 control how far a particle will move in a single iteration. Low values allow particles to roam far from target regions before being tugged back, while high values result in abrupt movement towards, or past, target regions. Typically, these are both set to a value of 2.0, although assigning different values to c_1 and c_2 sometimes leads to improved performance.

The basic PSO and its variants have successfully operated for continuous optimization functions. In order to extend the application to discrete space, Kennedy and Eberhart proposed a discrete binary version of PSO [24] where a particle moves in a state space restricted to zero and one on each dimension where each v_i represents the probability of bit s_i taking the value 1. Thus, the particles' trajectories are defined as the changes in the probability and v_i is a measure of individualćs current probability of taking 1. If the velocity is higher it is more likely to choose 1, and lower values favor choosing 0. A sigmoid function is applied to transform the velocity from real number space to probability space:

$$sig(v_{id}) = \frac{1}{1 + \exp(-v_{id})} \tag{3}$$

In the binary version of PSO, the velocities and positions of particles are updated as the following formulas:

$$v_{id}(t+1) = w v_{id}(t) + c_1 r \text{ and } 1(p_{id} - s_{id}(t)) + c_2 r \text{ and } 2(p_{gd} - s_{id}(t)) \tag{4}$$

$$s_{id}(t+1) = \begin{cases} 1, & \text{if } r \text{ and } 3 < sig(v_{id}) \\ 0, & \text{if } r \text{ and } 3 >= sig(v_{id}) \end{cases} \tag{5}$$

where s_{id} is the valued of the d-th dimension of particle s_i, and $s_{id} \in \{0, 1\}$; v_{id} is the corresponding velocity; $sig(v_{id})$ is calculated according to the Eq. 3, *rand*3 is a random number distributed in [0, 1]. As in basic PSO, a parameter $U_{\max}$ is incorporated to limit the v_{id} so that $sig(v_{id})$ does not approach too closely 0 or 1. Such implementation can ensure that the bit can transfer between 1 and 0 with a positive probability. In practice, $U_{\max}$ is often set at ± 4. The proposed algorithm is established based on standard PSO, namely basic PSO with inertia weight developed by Shi and Eberhart

in [53], where w is the inertia weight. The inertia weight controls the impact of previous histories of velocities on the current velocity, which is often used as a parameter to control the trade-off between exploration and exploitation. The particle adjusts its trajectory based on information about its previous best performance and the best performance of its neighbours. The inertia weight w is also used to control the convergence behaviour of the PSO. In order to reduce this weight over the iterations, allowing the algorithm to exploit some specific areas, the inertia weight w is updated according to the following equation:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \tag{6}$$

where $w_{\max}$, $w_{\min}$ are the maximum and minimum values that the inertia weight can take, and $iter$ is the current iteration (generation) of the algorithm while the $iter_{\max}$ is the maximum number of iterations (generations).

3.2 General Description of Hybrid Particle Swarm Optimization for the Location Routing Problem (HybPSO-LRP)

The proposed algorithm for the solution of the location routing problem, the hybrid particle swarm optimization (HybPSO-LRP), is a two phase algorithm (with an interaction between them) which combines a particle swarm optimization algorithm, the MPNS-GRASP algorithm [38], the expanding neighborhood search strategy [36] and a path relinking strategy [18]. In order to create the initial swarm of particles (each particle corresponds to a different solution), the MPNS-GRASP algorithm is used (see Section 3.3), where in the first (location) phase a capacitated facility location problem [13, 14] is solved in order to find the locations of the facilities and the assignments of the customers in the facilities and then in the second (routing) phase, for each facility, a vehicle routing problem [19, 56] is solved in order to find the routes.

Concerning the fitness function, it should be noted that in LRP, the fitness of each particle is related to the route length of each circle and to the set up cost of each facility, and since the problem that we deal with is a minimization problem, if a feasible solution has a high objective function value then it is characterized as an unpromising solution candidate. The expanding neighborhood search strategy (see Section 3.3) is utilized in order to improve the solutions of each particle in the swarm and it is used in order to reduce the computational time of the algorithm.

After the calculation of the initial population of the particles (solutions), we proceed to the main phase of the algorithm where the particles interact between them in order to achieve a better solution. In this phase of the algorithm the classic particle swarm optimization algorithm is combined with expanding neighborhood search and path relinking strategy (see Table 1). It should be noted that in the main phase of the proposed algorithm the location and routing problems are not solved separately but instead taking into account the particles of the initial phase and the equations used in the particle swarm optimization algorithm (see Section 3.1), each particle will move from its current solution to the global optimum (optimal solution of the whole swarm) or to the local optimum (optimal solution of each particle) changing each

Table 1 Hybrid particle swarm optimization algorithm for LRP

Initialization
Select the number of Swarms
Select the number of Particles for each swarm
Generate the initial population of the particles using MPNS-GRASP
Evaluate the fitness function of each particle
Apply expanding neighborhood search in each particle
Keep Optimum particle of the whole swarm
Keep Optimum solution of each particle
Main Phase
Do until the maximum number of generations has not been reached:
    Calculate the velocity of each particle
    Calculate the new position of each particle with path relinking
    Evaluate the new fitness function of each particle
    Apply expanding neighborhood search in each particle
    Update the optimum solution of each particle
    Update the optimum particle
    Update the inertia weight
Enddo
Return the best particle (the best solution)

time the appropriate elements in its corresponding array (these elements correspond to a routing and/or to a location decision). The particles usually in a discrete PSO will move from their current solution to the global optimum or to the local optimum by using the Eq. 4. However, in HybPSO-LRP, instead of this formula, a path relinking strategy (see Section 3.4) is used. Path relinking is an intensification Strategy that is used as a way of exploring trajectories between elite solutions. Thus, the current solution of each particle is combined using this strategy either with the global or the local optimum. In each iteration of the algorithm, the optimal solution of the whole swarm and the optimal solution of each particle are kept. A pseudocode of the proposed hybrid particle swarm optimization algorithm is presented in Table 1. In the following sections, an analytical presentation of the main steps of the HybPSO-LRP is given.

3.3 Initial Population

Instead of using a randomly generated initial population which may or may not necessarily contain good candidate solutions, a modified version of the well known greedy randomized adaptive search algorithm (GRASP), the multiple phase neighborhood search – GRASP (MPNS-GRASP) [38] is used to initialize the population. By the term, initial solution, we mean an initial location of the facilities and the assignments of the customers in the facilities and then, for each facility the construction of the initial routes.

GRASP [36, 50] is an iterative two phase search method which has gained considerable popularity in combinatorial optimization. Each iteration consists of two phases, a construction phase and a local search procedure. In the construction phase,

a randomized greedy function is used to build up an initial solution. This randomized technique provides a feasible solution within each iteration. This solution is, then, exposed for improvement attempts in the local search phase. The final result is simply the best solution found over all iterations.

That is, in the first phase, a *randomized greedy technique* provides feasible solutions incorporating both greedy and random characteristics. This phase can be described as a process which stepwise adds one element at a time to a partial (incomplete) solution. The choice of the next element to be added is determined by ordering all elements in a candidate list (the restricted candidate list – RCL) with respect to a greedy function. The heuristic is adaptive because the benefits associated with every element are updated during each iteration of the construction phase to reflect the changes brought on by the selection of the previous element. The probabilistic component of the *GRASP* is characterized by randomly choosing one of the best candidates in the list but not necessarily the top candidate. The greedy algorithm is a simple, one pass, procedure for solving the location routing problem. In the second phase, a *local search* is initialized from these points, and the final result is simply the best solution found over all searches.

The *multiple phase neighborhood search – greedy randomized adaptive search procedure – MPNS-GRASP* [38] is based on the design principles of the greedy randomized adaptive search procedure (GRASP). The most important features of MPNS-GRASP are:

- The use of a new construction scheme for the restricted candidate list (RCL), the ***cardinality based RCL construction scheme***. In most of the implementations of GRASP, some type of value based RCL construction scheme has been used. In such a scheme, an RCL parameter, α, determines the level of greediness or randomness in the construction. In our implementation the parameter α was not used and the best promising candidate samples (facilities in the location phase and customers in the routing phase) are selected to create the RCL. From this list, the first D samples (D is a parameter of the problem) are selected in order to form the final RCL. The candidate sample for inclusion in the solution is selected randomly from the RCL using a random number generator. Finally, the RCL is readjusted in every iteration by the recalculation of all the distances of the unassigned customers based on the new facilities and replacing the customer or facility which has been included in the solution by another customer or facility that does not belong to the RCL, namely the $(D + t_1)$th customer or facility where t_1 is the number of the current iteration.
- The *diversification of the greedy functions in each iteration of the MPNS-GRASP*, as it has the flexibility of applying alternative greedy functions in each iteration. Compared to most implementations of GRASP, where one simple greedy function is used for constructing the initial solution, the MPNS-GRASP uses a combination of greedy functions.
- MPNS-GRASP uses the *expanding neighborhood search* [36] method instead of a single local search method.
- In almost all implementations of GRASP the termination criterion is based on the maximum allowed number of iterations. Consequently, the algorithm wastes time in iterations that only add small, if any, improvements in the solution. In MPNS-GRASP, a termination criterion based on the *Lagrangean relaxation and subgradient optimization* [37] is used.

In MPNS-GRASP, the procedure expanding neighborhood search for the capacitated facility location problem is applied. *Expanding neighborhood search* (*ENS*) is a new metaheuristic algorithm used for the solution of combinatorial optimization problems and it has been proved very efficient for the solution of the travelling salesman problem [36] and the vehicle routing problem [39]. This metaheuristic algorithm overcomes a lot of obstacles that appear in heuristic and metaheuristic algorithms [35]. The most significant features of the *expanding neighborhood search – ENS* are:

- The *circle restricted local search moves strategy – CRLSM strategy*. This strategy is used in order to reduce the computational time of the algorithm by excluding from the search procedure all the edges that are not going to improve the solution. This happens by restricting the search into circles around the candidate for deletion edges [35, 36]. In the following, a description of the circle restricted local search moves strategy for a *2-opt trial move* [30] is presented. In this case, there are three possibilities based on the costs of the edges which are candidates for deletion and inclusion:
  - If both new edges increase in cost, a 2-opt trial move can not reduce the cost of the tour (e.g., in Fig. 1 [Possibility A], for both new edges the costs $C2$ and $C4$ are greater than the costs $B2$ and A of both old edges).
  - If one of the two new edges has cost greater than the sum of the costs of the two old edges, a 2-opt trial move again can not reduce the cost of the tour (e.g. in Fig. 1 [Possibility B] the cost of the new edge $C3$ is greater than the sum of the costs $A + B3$ of the old edges).
  - The only case for which a 2-opt trial move can reduce the cost of the tour is when at least one new edge has cost less than the cost of one of the two old edges (e.g., in Fig. 1 [Possibility C)], the cost $C1$ of the new edge is less than the cost of the old edge A) and the other edge has cost less than the sum of the costs of the two old edges (e.g., $C5 < A + B1$ in Fig. 1 [Possibility C]).

  Taking these observations into account, the circle restricted local search moves strategy restricts the search to edges where one of their end-nodes is inside a circle with radius length at most equal to the sum of the costs (lengths) of the two edges which are candidates for deletion.
- The *ability of the algorithm to change between different local search strategies inside the neighborhood search* is another feature of this algorithm. Sometimes, the use of one local search strategy leads to very limited search of the solution space and, thus, to only a small improvement of the solution. The idea of using a larger neighborhood to escape from a local minimum to a better one, had been proposed initially by Garfinkel and Nemhauser [15] and recently by Hansen and Mladenovic [20]. The expanding neighborhood search method starts with a local search strategy and with one specific size of the neighborhood search. If the solution is not improved satisfactorily, or it is not improved at all, with the use of the initial local search strategy, then before the expansion of the neighborhood search, a different local search strategy is chosen for exhausting all the possibilities of finding a better solution inside each radius of the circle of the neighborhood search. In the proposed algorithm, six different classic local search strategies are used. Four of them are used in the routing, namely the

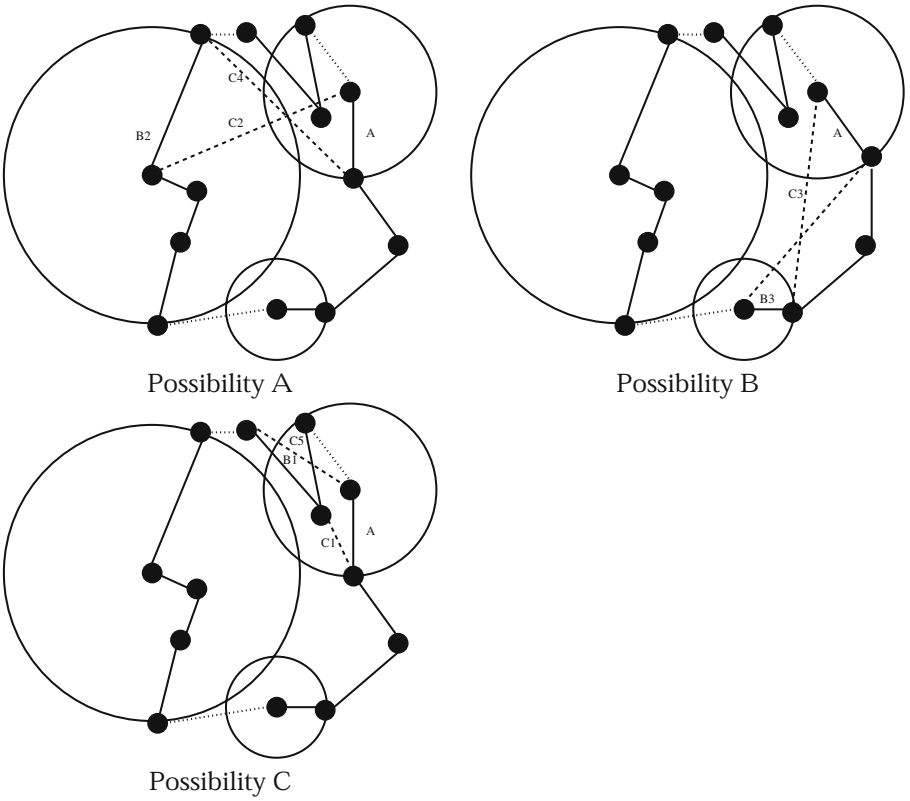

Fig. 1 Circle restricted local search moves strategy

2-opt, the 3-opt, the 1–1 exchange and the 1–0 relocate [39], and two of them are used in the location, namely the ADD and DROP heuristics [43].

- The problem of finding an appropriate stopping criterion that qualifies the solution found by the algorithm without wasting time in iterations that add small, if any, improvement to the solution is also addressed. The *ENS algorithm* suggests a simple but very effective solution to the problem. If, in addition to the solution of the problem, we calculate one lower bound of the solution by using the *Lagrangean relaxation and subgradient optimization* [37] for solving a relaxed problem, then, we can at any moment test the quality of the obtained solution. That is, first, a lower bound (LBD) and an upper bound (UBD) are calculated. Then, the parameter $e = \frac{UBD-LBD}{UBD}100\%$ is computed and if its value is less than a threshold value e_1, the algorithm stops with the current solution. Thus, without having any knowledge about the quality of the obtained solution, the algorithm stops when the difference between the lower bound and the upper bound (the current solution) is very small.
- The last feature of the expanding neighborhood search, is the use of an *expanding strategy* [36]. That is, initially, the search starts with a neighborhood that is created by the initially chosen circle of the circle restricted local search moves strategy. Inside this circle a number of different local search strategies,

depending on the decision (routing or location), are applied until all the possible trial moves have been explored and the solution can not further be improved in this neighborhood. Subsequently, the length of the radius of the circle is increased and, again, the same procedure is repeated until the stopping criterion is activated. Each different length of the neighborhood constitutes an external iteration. Initially, the size of the neighborhood, N_s, is defined based on the circle restricted local search moves strategy, for example $N_s = A/2$, where A is the cost of one of the candidates for deletion edges in the routing and the cost of the worst assignment customer in the location. For the selected size of the neighborhood, a number of different local search strategies are applied until all the possible trial moves have been explored and the solution can not further be improved in this neighborhood. The local search strategies are changed based on two conditions, first if the current local search strategy finds a local optimum and second if the quality of the current solution remains greater than the threshold number b_1 for a number of internal iterations. Subsequently, the quality of the current solution is compared with the current Lagrangean lower bound. If the quality of the solution is less than the threshold number e the algorithm stops, otherwise the neighborhood is expanded by increasing the length of the radius of the CRLSM strategy s by a percentage θ (e.g. $\theta = 10\%$), the Lagrangean lower bound is updated and the algorithm continues. In the routing, when the

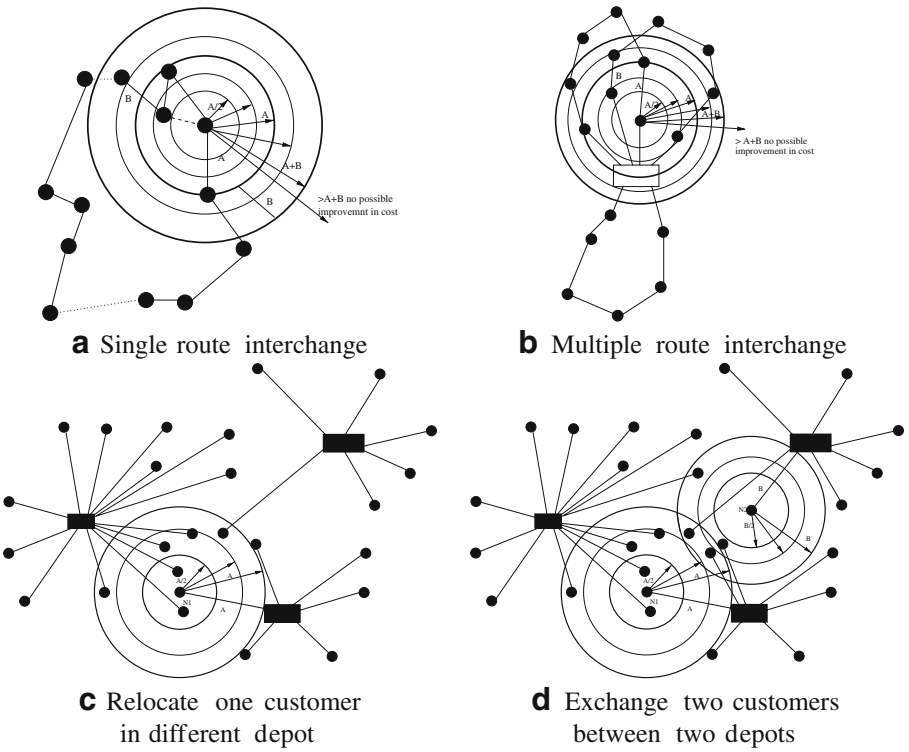

Fig. 2 Expanding neighborhood search strategy

length of the radius of the CRLSM strategy is equal to A, the length continues to increase until the length becomes equal to $A + B$, where B is the length of the other candidate for deletion edge. If the length of the radius of the CRLSM strategy is equal to $A + B$, and no stopping criterion has been already activated, then the algorithm terminates with the current solution. In Fig. 2a and b, the expanding neighborhood search method for the routing is presented. In contrast to the routing, in the location there are not exchanges based on the length of the edges that belonged in one or more routes. Thus, we have to modify slightly the basic part of the ENS algorithm, meaning the circle restricted local search moves strategy. Once an initial solution is at hand, we would like to find when there is a possibility for some of the local search strategies to be successfully applied to the problem. The length of the radius of the circles that will be created is equal to the length from the depot of the candidate for exchanging its assignment node. There are two different possibilities that the method examines for the particular problem. In the *first possibility* only one circle exists as shown in Fig. 2c. In this case, the examined node (customer) is the one with the worst assignment, meaning the node that it has the largest distance from the current assigned depot. If a better assignment is found then the new assignment is applied and the method continues for another node. The goal of each external iteration of the CRLSM Strategy is, now, the cost of the exchange of the assignment of the specific node or nodes to reduce the total assignment cost. The nodes that are very close to a candidate for exchange node and are assigned to different depots than the candidate for exchange node have more possibilities to change their assignment. The procedure stops when the radius of the circle becomes equal to the distance of the candidate for exchange node from the depot. The *second possibility* is to use two nodes that are assigned to different depots. The created circles around the two candidate nodes are examined. When two circles that are located around the two candidate nodes are intersected in two points, all the nodes that are located on the intersection space are examined in order to test the possibility of changing places taking into account the constraints in order to reduce the cost. The procedure stops when the radius of one of the two circles becomes equal to the distance of the candidate for exchange node from the depot. In Fig. 2d, it is observed that the search for a better assignment starts with the two nodes with the radius of the neighborhood search equal to $\frac{A}{2}$ for the one circle and equal to $\frac{B}{2}$ for the other. The two circles are expanded and are tested in parallel. For the nodes that belong in the circles and have initially assigned to a different depot, the possibility to have a better assignment with a use of a local search strategy is examined. In the intersection point of the two circles, if any, the search procedure changes. All the nodes that are inside the ellipse that is created from the two circles they are re-assigned taking into account their re-assignment not to violate the constraints and the procedure continues.

3.4 Path Relinking

This approach generates new solutions by exploring trajectories that connect high-quality solutions – by starting from one of these solutions, called the *starting solution* and generating a path in the neighborhood space that leads towards the other solution, called the *target solution* [18]. The roles of starting and target solutions

can be interchangeable. In the first one, the worst among the two solutions plays the role of the starting solution and the other plays the role of the target solution. In the second one, the roles are changing. There is a possibility for the two paths to simultaneously explored. A particle in particle swarm optimization can either follow its own way, or go back to its previous optimal solution, or go towards to the global optimal solution (to the best particle in the swarm). Thus, in the HybPSO-LRP when the particle decides to follow either the path to its previous optimal solution or the path to the global optimal solution, a path relinking strategy is applied where the current solution plays the role of the starting solution and the best particle of the swarm or the current best solution of the particle plays the role of the target solution. The trajectories between the two solutions are explored by simple swapping of two nodes of the starting solution if the decision corresponds to a routing or by closing a facility and opening another facility if the decision corresponds to a location until the starting solution becomes equal to the target solution. The paths are generated by choosing moves (swaps in the routing and opening or closing facilities in the location) in order to introduce attributes in the starting solution that are present in the guiding target solution. If in some step of the path relinking strategy a new best solution, either of the particle or of the whole swarm, is found then the current best (particle or swarm) solution is replaced with the new one and the algorithm continues.

4 Computational Results for the Location Routing Problem

The whole algorithmic approach was implemented in Fortran 90 and was compiled using the Lahey f95 compiler on a Centrino Mobile Intel Pentium M 750 at 1.86 GHz, running Suse Linux 9.1. The parameters of the proposed algorithm are selected after thorough testing and they are:

- Number of swarms: 1.
- Number of particles: 20.
- Number of generations: 100.
- $c_1 = 2, c_2 = 2$.
- $w_{\max} = 0.9, w_{\min} = 0.01$.
- Size of RCL = 50.
- $\theta = 10\%$.

It should be noted that the procedure for the selection of the most appropriate parameters started by using the parameters for PSO and GRASP that have been proposed in most papers in the literature. A number of tests were performed changing the initial parameters in order to succeed the best performance of the proposed algorithm. By this procedure the best performance was achieved by maintaining the initial values of the parameters $c_1, c_2, w_{\max}$ and $w_{\min}$ and by changing appropriately the initial values for all the other parameters. The algorithm was tested on a set of benchmark problems. It should be noted that there are not many papers in the literature that analyse and test the efficiency of the algorithms proposed in the past for the solution of location routing problems. Thus, a set of instances was created based on instances that most researchers have used (see [3] and http://sweet.ua.pt/~iscf143/_private/SergioBarretoHomePage.htm). The

Table 2 Benchmark instances' characteristics

No.	Names of researchers	Customers	Facilities	Vehicle capacity
1	Christofides and Eilon [9]	50	5	160
2	Christofides and Eilon [9]	75	10	140
3	Christofides and Eilon [9]	100	10	200
4	Daskin[11]	88	8	9,000,000
5	Daskin [11]	150	10	8,000,000
6	Gaskell [16]	21	5	6,000
7	Gaskell [16]	22	5	4,500
8	Gaskell [16]	29	5	4,500
9	Gaskell [16]	32	5	8,000
10	Gaskell [16]	32	5	11,000
11	Gaskell [16]	36	5	250
12	Min et al. [42]	27	5	2,500
13	Min et al. [42]	134	8	850
14	Perl and Daskin [47]	12	2	140
15	Perl and Daskin [47]	55	5	120
16	Perl and Daskin [47]	85	7	160
17	Perl and Daskin [47]	318	4	25,000
18	Perl and Daskin [47]	318	4	8,000
19	Or [45]	117	14	150

basic characteristics of the benchmark problems are given in Table 2. In this Table, the first column is the instance number, the second column shows the researcher that proposed each instance and the paper that the instance was, firstly, described. The third column shows the number of customers, the fourth column shows the number of facilities and the fifth column shows the vehicle capacity.

Table 3 presents the results of the HybPSO-LRP in the benchmark instances. In this Table, the first column is the instance number, the second column shows the solution given by the proposed HybPSO-LRP algorithm, the third column presents the best known solution (BKS) [3], the fourth column gives the quality of the solution of the proposed algorithm, while in the last column the computational time needed (in minutes) for finding the best solution by HybPSO-LRP is presented. The quality is given in terms of the relative deviation from the best known solution, that is $\omega = \frac{(c_{Hyb\,PSO-LRP} - c_{BKS})}{c_{BKS}}\%$, where $c_{Hyb\,PSO-LRP}$ denotes the cost of the solution found by HybPSO-LRP and c_{BKS} is the cost of the best known solution. It can be seen from Table 3, that the HybPSO-LRP algorithm, in 6 of the 19 instances has found a new best solution. For the rest of the instances, the quality of the solution is equal to the best known solution.

Table 4 presents the improvement of the solution given by HybPSO-LRP as the bounds (see Section 3.3) became more tight. Initially, we tested the algorithm with a threshold value e_1 equal to 5%. As it can be seen, in 9 of the 19 instances the algorithm has reached the best known solution. For the rest instances, the quality of the solution is between 0.02 and 0.73%. In the following, we gave a smaller value to the threshold number, i.e. e_1 is equal to 3%. We noticed in this case that the results were improved compared to the results taken when the threshold value was equal to 5%. We found new best solutions in six instances, in four instances we found

Table 3 Results of HybPSO-LRP in benchmark instances

No.	HybPSO-LRP	BKS	Quality (%)	CPU (min)
1	582.7	582.7	0.00	0.05
2	886.3	886.3	0.00	0.37
3	889.4	889.4	0.00	0.58
4	384.9	384.9	0.00	1.11
5	46,642.7	46,642.7	0.00	2.31
6	432.9	435.9	−0.68	0.01
7	588.5	591.5	−0.51	0.07
8	512.1	512.1	0.00	0.12
9	570.8	571.7	−0.15	0.15
10	511.1	511.4	−0.05	0.16
11	470.7	470.7	0.00	0.15
12	3,062	3,062	0.00	0.57
13	6,230	6,238	−0.12	1.24
14	204	204	0.00	0.01
15	1,135.9	1,136.2	−0.02	0.27
16	1,656.9	1656.9	0.00	0.57
17	580,680.2	580,680.2	0.00	3.18
18	747,619	747,619	0.00	3.10
19	12,474.2	12,474.2	0.00	1.32

the current best known solution and for the rest of the instances, the quality of the solution is between 0.01 and 0.26%. Finally, the algorithm was tested with threshold value equal to 1%. Now, in 6 of the 19 instances the algorithm has improved the new best solution that has already been found by the algorithm when the threshold value was equal to 3%. For the rest of the instances, the quality of the solution is equal to the best known solution. It should be mentioned that if the maximum number of generations has been reached and the value of the quantity e is still remaining greater than e_1 the algorithm stops with the current best particle (solution).

In order to give the significance of each of the characteristics (metaheuristics used) of the HybPSO-LRP, we implement a number of different versions of a particle swarm optimization Algorithm for LRP. In these implementations, the basic characteristics of the HybPSO-LRP are not included in order to prove the contribution of each of the characteristics of the proposed algorithm. More precisely, initially a particle swarm optimization algorithm is tested without the MPNS-GRASP, the expanding neighborhood search strategy and the path relinking strategy (columns 2 and 3 of Table 5), afterwards the MPNS-GRASP strategy (columns 4 and 5 of Table 5), and the ENS Strategy (columns 6 and 7 of Table 5) are added. Finally, in the last two columns of the Table 5 the results of the HybPSO-LRP are presented when the path relinking strategy is added. In the implementations where the path relinking strategy is not included the Eq. 4 is used in order for the particle to decide if will follow its own way or go back to its previous optimal solution, or go towards to the global optimal solution (to the best particle in the swarm). In all implementations, the parameters were set equal to the selected parameters of HybPSO-LRP and the local search strategies were the same as in HybPSO-LRP so as the algorithms to run under the same circumstances and thus their results to be comparable. In Table 5, the cost and the computational time of all implementations are presented. From this

Table 4 Comparison of different threshold values

No.	$e_1 = 5\%$		$e_1 = 3\%$		$e_1 = 1\%$	
	Solution	Quality (%)	Solution	Quality (%)	Solution	Quality (%)
1	582.7	0.00	582.7	0.00	582.7	0.00
2	891.3	0.56	887.8	0.17	886.3	0.00
3	895.9	0.73	891.8	0.26	889.4	0.00
4	386.1	0.31	385.8	0.23	384.9	0.00
5	46,686.12	0.09	4,6670.35	0.05	46,642.7	0.00
6	435.9	0.00	435.1	−0.18	432.9	−0.68
7	591.5	0.00	590.9	−0.10	588.5	−0.51
8	512.1	0.00	512.1	0.00	512.1	0.00
9	571.7	0.00	571.1	−0.10	570.8	−0.15
10	511.4	0.00	511.2	−0.03	511.1	−0.05
11	472.1	0.29	471.8	0.23	470.7	0.00
12	3,062	0.00	3,062	0.00	3,062	0.00
13	6,242.1	0.06	6,234	−0.06	6230	−0.12
14	204	0.00	204	0.00	204	0.00
15	1,136.2	0.00	1,135.9	−0.02	1,135.9	−0.02
16	1,661.34	0.26	1,657.2	0.01	1,656.9	0.00
17	581,478.35	0.13	581,223.46	0.09	580,680.2	0.00
18	747,834.55	0.02	747,723.12	0.01	747,619	0.00
19	12,526.34	0.41	12,498.9	0.19	12,474.2	0.00

Table 5 Comparison of the proposed algorithm with other PSO implementations

	PSO		PSO-MPNS-GRASP		PSO-MPNS-GRASP-ENS		HybPSO-LRP	
	Cost	CPU (min)	Cost	CPU (min)	Cost	CPU (min)	Cost	CPU (min)
1	582.7	0.98	582.7	0.87	582.7	0.07	582.7	0.06
2	888.9	1.21	887.1	1.12	886.9	0.35	886.3	0.37
3	895.7	1.15	893.2	1.15	891.5	0.64	889.4	0.58
4	386.1	2.31	385.3	2.21	385.2	1.12	384.9	1.11
5	46,657.3	4.28	46,645.2	4.01	46,644.1	2.51	46,642.7	2.31
6	437.1	1.01	435.9	0.67	435.9	0.02	432.9	0.01
7	592.1	1.12	591.8	0.55	591.7	0.07	588.5	0.07
8	512.1	0.86	512.1	0.81	512.1	0.15	512.1	0.12
9	574.1	1.01	571.7	0.98	571.7	0.18	570.8	0.15
10	512.1	1.12	511.4	1.23	511.1	0.23	511.1	0.16
11	470.7	1.23	470.7	1.32	470.7	0.28	470.7	0.15
12	3,078	1.17	3,062	1.31	3,062	0.51	3,062	0.57
13	6,241	3.48	6,230	3.51	6,230	1.18	6,230	1.24
14	204	0.37	204	0.24	204	0.01	204	0.01
15	1,137.5	1.32	1,136.2	1.27	1,135.9	0.45	1,135.9	0.27
16	1,658.4	2.12	1,657.1	2.01	1,656.9	0.67	1,656.9	0.57
17	580,701.4	6.95	580,692.4	6.89	580,691.5	3.15	580,680.2	3.18
18	747,679	6.30	747,656	6.56	747,632	3.21	747,619	3.10
19	12,481.2	3.14	12477.1	3.24	12,476.7	1.57	12,474.2	1.32

Table, it can be observed that the use of each of the characteristics in the HybPSO-LRP improves significantly either the quality of the solution or the computational time or both of them. More precisely, the addition of the MPNS-GRASP in the initial PSO algorithm gave much better results regarding the quality of the solution but the computational time was not improved at all. The computational time was improved significantly with the addition of the expanding neighborhood search strategy, but the results were almost the same as in the previous case. The significant improvement in the quality of the solutions was achieved with the addition of the path relinking strategy. The reason is that now the particles move in a faster and more efficient way to their local optimum or to the global optimum solution (to the best particle in the swarm). It can, also, be observed that in some instances, namely for instances 1, 8, 11 and 14, all four implementations found the same solutions but the significant difference is the time that they needed to find this solution. More precisely, while in the first implementation the computational time to find the optimum is equal to 0.98 min and, in the second implementation, it is equal to 0.87 min, the addition of the ENS reduced the computational time (third implementation) to 0.07 min and the addition of the path relinking (HybPSO-LRP) algorithm reduced even more the computational time to find the optimum to 0.06 min. It should be, also, noted that the characteristic that led to a new best solution it is different in some instances. That is, in instances 6, 7 and 9 the addition of the path relinking component of the algorithm improved the solution and gave the new best solution. On the other hand, for the instance 10 and 15 the addition of the expanding neighborhood search strategy gave the required flexibility to give the new best solution. Finally, for the instance 13 the most important component is the MPNS-GRASP where with its addition a new best solution was found.

5 Conclusions

In this paper, a nature inspired approach has been introduced for the effective handling and optimization of logistics management problems. Specifically, a hybrid algorithmic nature inspired methodology was proposed, namely the HybPSO-LRP algorithm, for the effective handling of the location routing problem. This algorithm is a general algorithm that can be applied with remarkable results both to quality and computational efficiency to many combinatorial optimization and supply chain problems. One of the main contributions of this paper is to show that the particle swarm optimization can be used in hybrid synthesis with other metaheuristics for the solution of the location routing problem. A second contribution is the utilization of the MPNS-GRASP procedure for the generation of the initial particles. One of the main problems that one has to deal with is how the particles will move from their current solution to the global optimum (optimal solution of the whole swarm) or to the local optimum (optimal solution of each particle). HybPSO-LRP uses the path relinking strategy instead of the classic way that usually the particles move from their current solution to the local optimum or to the global optimum. Finally, the expanding neighborhood search strategy is utilized in order to improve the solutions of each particle in the swarm. The algorithm was tested on a set of 19 benchmark instances. The algorithm was thoroughly tested with different parameter values. The algorithm has found new best solutions in 6 of the 19 instances and for the rest of the

instances the quality of the solution is equal to the best known solution. Thus, these tests proved the computational efficiency of the proposed algorithm.

References

1. Albareda-Sambola, M., Diaz, J.A., Fernandez, E.: A compact model and tight bounds for a combined location-routing problem. Comput. Oper. Res. **32**(3), 407–428 (2005)
2. Averbakh, I., Berman, O.: Routing and location-routing p-delivery men problems on a path. Transp. Sci. **28**(2), 162–166 (1994)
3. Barreto, S., Ferreira, C., Paixao, J., Santos, B.S.: Using clustering analysis in a capacitated location-routing problem. Eur. J. Oper. Res. **179**(3), 968–977 (2007)
4. Bookbinder, J.H., Reece, K.E.: Vehicle routing considerations in distribution system design. Eur. J. Oper. Res. **37**, 204–213 (1988)
5. Caballero, R., Gonzalez, M., Guerrero, F.M., Molina, J., Paralera, C.: Solving a multiobjective location routing problem with a metaheuristic based on tabu search. Application to a real case in Andalusia. Eur. J. Oper. Res. **177**(3), 1751–1763 (2007)
6. Cappanera, P., Gallo, G., Maffioli, F.: Discrete facility location and routing of obnoxious activities. Discrete Appl. Math. **133**(1–3), 3–28 (2003)
7. Chan, Y., Baker, S.F.: The multiple depot, multiple traveling salesmen facility-location problem: vehicle range, service frequency, and heuristic implementations. Math. Comput. Model. **41**(8–9), 1035-1053 (2005)
8. Chiang, W.C., Russell, R.A.: Integrating purchasing and routing in a propane gas supply chain. Eur J. Oper. Res. **154**(3), 710–729 (2004)
9. Christofides, N., Eilon, S.: An algorithm for the vehicle dispatching problem. Oper. Res. Q. **20**, 309–318 (1969)
10. Christofides, N., Eilon, S.: Expected distances for distribution problems. Oper. Res. Q. **20**, 437–443 (1969)
11. Daskin, M.: Network and Discrete Location. Models, Algorithms and Applications. Wiley, New York (1995)
12. Dondo, R., Cerda, J.: A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. Eur. J. Oper. Res. **176**(3), 1478–1507 (2007)
13. Franchis, R.L., White, J.H.: Facility Layout and Location. An Analytic Approach, Prentice-Hall, New Jersey (1974)
14. Franchis, R.L., McGinnis, L.F., White, J.A.: Locational analysis. Eur. J. Oper. Res. **12**(3), 220–252 (1983)
15. Garfinkel, R., Nemhauser, G.: Integer Programming. Wiley, New York (1972)
16. Gaskell, T.J.: Bases for vehicle fleet scheduling. Oper. Res. Q. **18**, 281–295 (1967)
17. Ghosh, J.K., Sinha, S.B., Acharya, D.: A generalized reduced gradient based approach to round-trip location problem. In: Jaiswal, N.K. (ed.) Scientific Management of Transport Systems, pp. 209–213. Amsterdam, Holland (1981)
18. Glover, F., Laguna, M., Marti, R.: Scatter search and path relinking: advances and applications. In: Glover, F., Kochenberger, G.A. (eds.) Handbook of Metaheuristics, pp. 1–36. Kluwer, Boston (2003)
19. Golden B.L., Assad, A.A.: Vehicle Routing: Methods and Studies. North Holland, Amsterdam (1988)
20. Hansen, P., Mladenovic, N.: Variable neighborhood search: principles and applications. Eur. J. Oper. Res. **130**, 449–467 (2001)
21. Hwang, H.S.: Design of supply-chain logistics system considering service level. Comput. Ind. Eng. **43**(1–2), 283–297 (2002)
22. Jacobsen, S.K., Madsen, O.B.G.: A comparative study of heuristics for two level routing location problem. Eur. J. Oper. Res. **5**, 378–387 (1980)
23. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of 1995 IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
24. Kennedy, J., Eberhart, R.: A discrete binary version of the particle swarm algorithm. In: Proceedings of 1997 IEEE International Conference on Systems, Man, and Cybernetics, vol. 5, pp. 4104–4108 (1997)

25. Laoutaris, N., Zissimopoulos, V., Stavrakakis, I.: On the optimization of storage capacity allocation for content distribution. Comput. Networks **47**(3), 409–428 (2005)
26. Laporte, G.: Location routing problems. Golden, B.L., et al. (eds.) Vehicle Routing: Methods and Studies, pp. 163–198. North Holland, Amsterdam (1988)
27. Laporte, G., Dejax, P.J.: Dynamic location-routing problems. J. Oper. Res. Soc. **40**(5), 471–482 (1989)
28. Laporte, G., Nobert, Y., Arpin, D.: An exact algorithm for solving a capacitated location-routing problem. Ann. Oper. Res. **6**, 293–310 (1986)
29. Laporte, G., Nobert, Y., Pelletier, P.: Hamiltonian location problems. Eur. J. Oper. Res. **12**(1), 82–89 (1983)
30. Lin, S.: Computer solutions of the traveling salesman problem. Bell Syst. Tech. J. **44**, 2245–2269 (1965)
31. Lin, C.K.Y., Chow, C.K., Chen, A.: A location-routing-loading problem for bill delivery services. Comput. Ind. Eng. **43**(1–2), 5–25 (2002)
32. Lin, C.K.Y., Kwok, R.C.W.: Multi-objective metaheuristics for a location-routing problem with multiple use of vehicles on real data and simulated data. Eur. J. Oper. Res. **175**(3), 1833–1849 (2006)
33. Liu, S.C., Lee, S.B.: A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration. Int. J. Adv. Manuf. Technol. **22**(11–12), 941–950 (2003)
34. Madsen, O.B.G.: Methods for solving combined two level location routing problems of realistic dimension. Eur. J. Oper. Res. **12**(3), 295–301 (1983)
35. Marinakis, Y.: Vehicle routing in distribution problems. Ph.D. thesis, Technical University of Crete, Department of Production Engineering and Management, Chania, Greece (2005)
36. Marinakis, Y., Migdalas, A., Pardalos, P.M.: Expanding neighborhood grasp for the traveling salesman problem. Comput. Optim. Appl. **32**, 231–257 (2004)
37. Marinakis, Y., Migdalas, A., Pardalos, P.M.: A hybrid genetic–GRASP algorithm using lagrangean relaxation for the traveling salesman problem. J. Comb. Optim. **10**, 311–326 (2005)
38. Marinakis, Y., Migdalas, A., Pardalos, P.M.: Multiple phase neighborhood search GRASP based on Lagrangean relaxation and random backtracking Lin–Kernighan for the traveling salesman problem. J. Comb. Optim. **38**, 555–580 (2006)
39. Marinakis, Y., Marinaki, M., Migdalas, A.: A hybrid Genetic–GRASP–ENS algorithm for the vehicle routing problem. IEEE Trans. Evol. Comput. (2007) (submitted)
40. Melechovsky, J., Prins, C., Calvo, R.W.: A metaheuristic to solve a location-routing problem with non-linear costs. J. Heuristics **11**(5–6), 375–391 (2005)
41. Min, H.: Consolidation terminal location-allocation and consolidated routing problems. J. Bus. Logist. **17**(2), 235–263 (1996)
42. Min, H., Jayaraman, V., Srivastava, R.: Combined location-routing problems: a synthesis and future research directions. Eur. J. Oper. Res. **108**, 1–15 (1998)
43. Nagy, G., Salhi, S.: Nested heuristic methods for the location-routing problem. J. Oper. Res. Soc. **47**, 1166–1174 (1996)
44. Nagy, G., Salhi, S.: Location-routing: issues, models and methods. Eur. J. Oper. Res. **177**, 649–672 (2007)
45. Or, I.: Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking. Ph.D. thesis, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL (1976)
46. Perl, J., Daskin, M.S.: A unified warehouse location-routing methodology. J. Bus. Logist. **5**(1), 92–111 (1984)
47. Perl, J., Daskin, M.S.: A warehouse location routing model. Transp. Res. B **19**, 381–396 (1985)
48. Prins, C., Prodhon, C., Calvo, R.W.: Solving the capacitated location-routing problem by a grasp complemented by a learning process and a path relinking. 4OR **4**, 221–238 (2006)
49. Prins, C., Prodhon, C., Calvo, R.W.: A memetic algorithm with population management (MA|PM) for the capacitated location-routing problem. In: Evolutionary Computation Combinatorial Optimization. LNCS, vol. 906, pp. 183–194 (2006)
50. Resende, M.G.C., Ribeiro, C.C.: Greedy randomized adaptive search procedures. In: Glover, F., Kochenberger, G.A. (eds.) Handbook of Metaheuristics. Kluwer, Boston, pp. 219–249 (1998)
51. Russell, R., Chiang, W.C., Zepeda, D.: Integrating multi-product production and distribution in newspaper logistics. Comput. Oper. Res. (2007) doi:10.1016/j.cor.2006.09.002
52. Simchi-Levi, D., Berman, O.: A heuristic algorithm for the traveling salesman location problem on networks. Eur. J. Oper. Res. **36**, 478–484 (1988)

53. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of 1998 IEEE World Congress on Computational Intelligence, pp. 69–73 (1998)
54. Srivastava, R., Benton, W.C.: The location-routing problem: consideration in physical distribution system design. Comput. Oper. Res. **6**, 427–435 (1990)
55. Stowers, C.L., Palekar, U.S.: Location models with routing considerations for a single obnoxious facility. Transp. Sci. **27**(4), 350–362 (1993)
56. Toth, P., Vigo, D.: The Vehicle Routing Problem, Monographs on Discrete Mathematics and Applications, Siam (2002)
57. Tuzun, D., Burke, L.I.: A two-phase tabu search approach to the location routing problem. Eur. J. Oper. Res. 116, 87–99 (1999)
58. Watson-Gandy, C.T.D., Dohrn, P.J.: Depot location with van salesman – a practical approach. Omega **1**, 321–329 (1973)
59. Wu, T.H., Low, C., Bai, J.W.: Heuristic solutions to multi-depot location-routing problems. Comput. Oper. Res. **29**, 1393–1415 (2002)
60. Zografos, K.G., Samara, S.: Combined location-routing model for hazardous waste transportation and disposal. Transp. Res. Record **1245**, 52–59 (1989)