Bio-inspired Metaheuristics for the Vehicle Routing Problem

DANIELA PONCE
Department of Information Technologies, Faculty of Informatics and Management
University of Hradec Králové
Rokitanského 62, 500 02 Hradec Králové
CZECH REPUBLIC
daniela.ponce@uhk.cz http://www.uhk.cz

Abstract: - The paper proposes a metaheuristic algorithm based on particle swarm optimization (PSO) for the problem of vehicle routing (VRP). A direct sequence-based representation of VRP solution is defined, i.e. each particle represents a set of feasible vehicle routes. After initial routes have been constructed, a modified edge recombination crossover operator is used to move particles towards better solutions. The paper presents early results on application of the proposed PSO algorithm to VRP.

Key-Words: - metaheuristic algorithm, vehicle routing problem, particle swarm optimization, edge recombination crossover operator

1 Introduction

In the vehicle routing problem (VRP), a set of routes has to be specified such that vehicles running on the routes will serve all customers, while all conditions regarding the customers, vehicles and routes are hold and the set is optimal.

Since the VRP formulation fifty years ago, many problem variants have been defined ([7]). In this paper, we deal with the so-called capacitated vehicle routing problem (CVRP): the customers have deterministic delivery demand only, the vehicles have equal capacities and they all start and end at a single depot. The total route cost is subject to optimization and it is only composed of distance-dependent travel costs of vehicles.

Because the VRP is an important NP-hard optimization problem plenty of exact, heuristic and metaheuristic solution techniques have been invented. In our contribution, we present a metaheuristic solution technique based on particle swarm optimization (PSO). PSO is a population-based algorithm is inspired by spatial behavior of large groups of animals (e.g. bird flocks or fish schools) which are capable to move in a coordinated manner towards common target without central control. Although PSO has been used for many optimization problems, few applications for VRP are referenced ([5]). Recently, PSO for VRP with simultaneous pickup and delivery has been presented ([A]).

The paper is organized in the following way. First, we define the problem of capacitated vehicle routing and we mention some solution methods. Then, we introduce the particle swarm optimization and we propose its application for CVRP including the explanation of the modified edge recombination operator. Finally, we discuss the results and future work.

2 Capacitated Vehicle Routing Problem

We can define the capacitated vehicle routing problem (CVRP) formally in the following way. Let G = (V, A) denote a graph such that $V = \{v_0, v_1, \ldots, v_n\}$ is a set of vertices with $v_1, \ldots, v_n$ vertices representing customers and v_0 denoting a depot, and $A = \{(v_i, v_j) | v_i, v_j \in V, i \neq j\}$ is a set of arcs. Let and let k_{max} denote the number of vehicles available, all stationed in the depot v_0 and all with the same capacity Q. For each customer, delivery demand quantity q_i is defined. Finally, a matrix of travel costs associated with arcs (c_{ij}) for each $(v_i, v_j) \in A$ is also defined.

A solution of CVRP is defined as a set of at most k routes keeping the following conditions:

each route starts and terminates at the depot, (1)
each customer is visited exactly once by exactly one vehicle, (2)
the total demand of customers in each route does not exceed the vehicle capacity. (3)

Solution of CVRP is subject to optimization. Let the total cost of all routes be defined as the sum of travel costs spent by vehicles running on arcs that are parts of respective routes. The following condition has to be fulfilled, too:

the total cost of all routes is minimized. (4)

2.1 Metaheuristic Algorithms for CVRP

At first CVRP was tackled by exact methods which can be divided into three big groups: branch-and-bound algorithms, branch-and-cut algorithms and set-covering-based algorithms ([7]). Later on, better performing heuristic methods were formulated. The *constructive heuristics* incrementally constructs feasible solutions while the cost of the solution is controlled. In *two-phase heuristics*, the solution is found in two phases: clustering of vertices into groups to be served by individual vehicles and ordering vertices into actual vehicle routes. The order of the two phases is not given: cluster-first, route-second methods first cluster the vertices and then arrange them into route, while route-first, cluster-second methods first order the vertices into one megaroute and then divide it into feasible routes. The *improvement heuristics* gradually modify feasible solutions by exchanging nodes or edges within or between the vehicle routes. ([4]).

Recently, new kind heuristic techniques have been devised, coined metaheuristics. Typically, the solutions generated during the solution search process may be of poorer quality or even infeasible ([3]). These methods focus on deep exploration of the most promising regions of the solution space ([4]). The main metaheuristics widely used for the CVRP are simulated annealing, deterministic annealing, tabu search, genetic algorithms, ant algorithms and neural networks ([3]). Very recently, particle swarm optimization and artificial immune systems have been used as yet another two promising metaheuristic algorithms for solving CVRP, however, still few results can be found in the literature ([6]).

3 Particle Swarm Optimization

PSO is a population-based algorithm ([2]) that simulates spatial behavior of animals living in groups such as flock of birds or school of fish in which the individuals move toward common target without central control. The group is called swarm and the individuals are called particles. Each particle represents one solution of the problem. For each particle, position in the solution space and velocity driving its movement in the solution space is specified.

Let $\mathbf{x}_i(t)$ denote the position of the ith particle in the solution space at time step t and let $\mathbf{v}_i(t)$ denote the velocity vector of the ith particle in the solution space at time step t. The new position of the ith particle is calculated as:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (5)$$

The velocity vector comprises the individual experience of the particle (so-called cognitive component) as well as the collective experience of the swarm (so-called social component). Each particle keeps information about the best position it has ever reached as personal best position. For each position, its quality is calculated using the fitness function defined on the solution space.

For each particle, its neighbors are specified according to the neighborhood structure defined. One of the basic neighborhood structures is so-called total star neighborhood where each particle is a neighbor of all other particles. Each particle also keeps information about the best position ever reached by its neighbors as global best position.

Let $\mathbf{y}_i(t)$ denote the personal best position of the ith particle at time step t, and let $\hat{\mathbf{y}}_i(t)$ denote the global best position of the ith particle at time step t. The velocity of the ith particle is calculated as:

$$v_{ij}(t+1)=v_{ij}(t)+c_1r_{1j}(t)[y_{ij}(t)-x_{ij}(t)]+c_2r_{2j}(t)[\hat{y}_{ij}(t)-x_{ij}(t)] \quad (6)$$

so the new position of a particle can be formulated as:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t)+c_1r_{1j}(t)[y_{ij}(t)-x_{ij}(t)]+ +c_2r_{2j}(t)[\hat{y}_{ij}(t)-x_{ij}(t)] \quad (7)$$

Cognitive and social acceleration coefficients c_1 and c_2 serve the purpose to strengthen or weaken the attraction of a particle towards its personal best and global best positions, respectively. They are usually constant during the iterations. Random coefficients $\mathbf{r}_1$ and $\mathbf{r}_2$ in the range <0;1> represent small perturbations in the attraction of a particle towards the personal best and global best positions, respectively. Typically, the c_1 and c_2 constants are set to a value in the range <0;2>, the swarm size is between 20 to 50, and the simulation is run from 500 to 5000 time steps. The basic version of the PSO is given in as Algorithm 1.

Algorithm 1. Basic PSO
create and initialize the swarm of N particles;
repeat
    for each particle p **do** {
        if f(p.x) < f(p.y) **then** p.y = p.x;
        for each neighbor q of particle p **do** {
            if f(q.x) < f(p.x) **then** p.ŷ = q.x;}}
    for each particle p **do** {
        update the position p.x using equation 7;}
until stopping criterion is true;}

3.2 PSO Representation of CVRP

CVRP solution can be represented in PSO using different coding schemas. Ai and Kachitvichyanukul represented the CVRP solution as priorities assigned to customers ([1]). The solution is used in the constructive heuristics: the higher the priority, the earlier the customer will be placed on a route to be served. We propose a sequence based representation: the particle represents a collection of vehicle routes.

Let n denote the number of customers and k the number of vehicles used, $k \leq k_{max}$. The number of vehicles used selected in such a way that:

$$(k-1)*Q < \sum_{i=1}^{n} q_i \leq k*Q \tag{8}$$

Each particle represents a set of k routes. The position **x** of a particle is defined as n+k-1 dimensional vector such that:

$(\mathbf{x})_i \in \{0, 1, \ldots, n\}$ for $0 \leq i \leq n+k-1$, where (9)
$(\mathbf{x})_i = j$ for exactly one value of $i, 1 \leq j \leq n$, and (10)
$(\mathbf{x})_i = 0$ for exactly (k-1) values of i. (11)

$a=(\mathbf{x})_i$ and $b=(\mathbf{x})_{i+1}$ adjacent values in the position represent the fact that the arc (v_a, v_b) is included in the route. The arc (v_0, v_b) represents the starting arc of the route of a vehicle, the arc (v_a, v_0) represents the ending arc of the route of a vehicle. For $c=(\mathbf{x})_1$ and $d=(\mathbf{x})_{n+k-1}$, the (v_0, v_c) and (v_d, v_0) arcs are also added at the beginning and the end of the corresponding routes, respectively. For example, the position **x**=(8,17,11,12, 15,0,6,16,2,4,18,0,5,1,9,13,3,10,7,14) represents three routes on customers $\{v_1, \ldots, v_{18}\}$ (see Fig. 1).

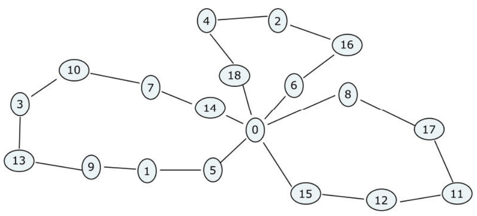

Fig. 1. Three routes on customers $\{v_1, \ldots, v_{18}\}$.

The representation guarantees that the solution encoded by the position of a particle will hold the conditions (1) and (2) of the optimization problem. The condition (3) of the optimization problem is kept during the process of the particle construction. During the solution search process, when the particles move in the solution space, the position is adjusted if the condition (3) does not hold. The fitness function includes a penalty function which adds a big penalty to the resulting fitness of the solution if the (3) condition is violated. The fitness function is formulated as total travel costs of the route.

Since the representation of solution is sequence-based, the general PSO model has to be reinterpreted. Subtraction and addition of positions is not defined, therefore we apply edge recombination crossover (ERX) operator on position pairs. The result of such an operation is again a new position such that it shares common parts with both operand positions. We reformulate the calculation of the new position of a particle in the following way:

$$\mathbf{x}_i(t+1) = \begin{cases} \mathbf{y}_i(t) \otimes \mathbf{x}_i(t), \text{if } p_i(t) \in \left\langle \dfrac{u_i(t)}{u_i(t)+\hat{u}_i(t)}; 1 \right\rangle \\ \hat{\mathbf{y}}_i(t) \otimes \mathbf{x}_i(t), \text{if } p_i(t) \in \left\langle 0; \dfrac{u_i(t)}{u_i(t)+\hat{u}_i(t)} \right\rangle, \end{cases} \tag{12}$$

where $\otimes$ denotes the edge recombination crossover operator, $p_i(t) \in <0;1>$ is assigned a value with uniform distribution probability, $u_i(t)$ is fitness of $\mathbf{y}_i(t)$ and $\hat{u}_i(t)$ is fitness of $\hat{\mathbf{y}}_i(t)$.

3.3 Edge Recombination Crossover Operator

Originally, the edge recombination crossover operator was used as crossover operator in genetic algorithm for the travelling salesman problem ([8]). The ERX can be used if the solution of TSP is represented as a path. The ERX operator retains the path segments used in one or other parent solution. When applying the ERX operator to two operand (parent) solutions (e.g. see Fig. 2), we get a new offspring solution such that each segment of the offspring solution comes from one or other parent solutions (e.g. see Fig. 3).

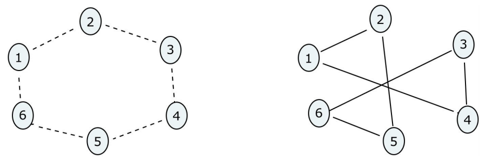

Fig. 2. Two parent paths with shared segments.

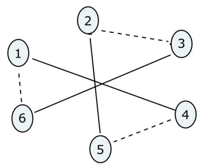

Fig. 3. Offspring path.

The application of the ERX operator is performed in the following way. Firstly, the adjacency matrix of all nodes contained in the both operand paths is constructed in such a way that for every node the nodes adjacent with the given node in either operand paths are listed. Duplicate nodes on the list of adjacent nodes mean that the segment is common for the both operand paths.

The adjacency matrix is used in the resulting path construction. Initially, a starting node n_s of the path and one node n_a from the list of its adjacent nodes are selected and they become the first segment of the resulting path.

Adjacency matrix	
node	adjacent nodes
1	~~2~~,6,~~2~~,~~4~~
2	~~1~~,3,~~1~~,5
3	~~2~~,~~4~~,~~4~~,6
4	3,5,~~1~~,3
5	~~4~~,6,~~2~~,6
6	~~1~~,5,3,5

Resulting path	
node	adjacent nodes
1	2,4
2	1
4,	1,5
5	4

Tab. 1 Construction of the resulting path, n_s=4,n_a=5.

The rest of the path is constructed in a loop: the information on adjacency of n_s and n_a in the resulting path is recorded for both nodes n_s and n_a. Furthermore, the node n_s is removed from the list of adjacent nodes for all nodes in the adjacency matrix. The node n_a gets then assigned to n_s, and n_a is assigned a node selected from the list of the adjacent nodes of n_s (see Tab. 1). The loop terminates when the list of the adjacent nodes of the current n_s becomes empty and the adjacency matrix is empty. The last node of the path is then made adjacent with the node which was used to start the path.

3.4 ERX Operator for Vehicle Routes

In the case of CVRP, usually we deal with several routes which all start and end in the depot (see e.g. Fig. 4).

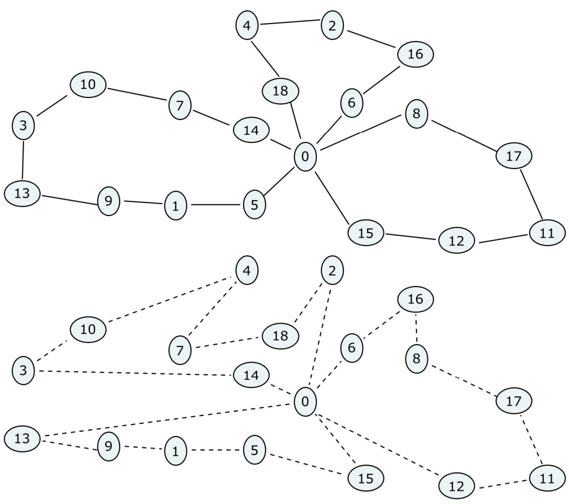

Fig. 4. Two operand route sets.

During the construction of the resulting route set, we have to deal differently with the node 0 in the role of n_s: it cannot be removed from the list of adjacent nodes for all nodes in the adjacency matrix, as the construction would result in a single huge route passing through the depot node and all nodes.

On the other hand, ignoring the node 0 in the adjacency matrix, routes can be closed too early leaving some nodes not included in any route. Therefore, when selecting the next n_a node from the adjacent nodes of the current n_s node, we give less priority to the node 0 than to other nodes. We also give higher priority to duplicate adjacent nodes of the current n_s node (meaning common segments in the operand solutions). When the list of the adjacent nodes of the current n_s is empty, we close the current route, and we start another one by selecting such a node in the adjacency matrix that still has adjacent nodes listed. The application of the ERX operator halts when the adjacency matrix becomes empty.

3.5 Refinement of the Solution

In order to finish the construction of new solution, we have to make sure that the solution is composed of k routes and that each route passes through the depot. If necessary (see e.g. Fig. 5), we randomly merge two routes into one or split one route into two until we get k number of routes such that they all pass through the depot.

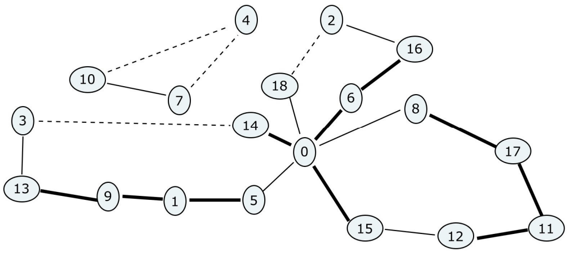

Fig. 5. The result of the ERX operator application on the operand route sets from Fig. 4.

Finally, when k number of routes is reached, we have to verify that the condition (3) is met. We can apply some multiroute improvement heuristic regarding the edge exchange (see [4] for details) to balance the total demand of routes.

4 Results and Future Work

We have run the algorithm proposed on several CVRP instances. As little problem-specific information was used to guide the movement of the particle swarm in the solution space, the results are not competitive with the other results achieved.

In the currently developed algorithm, the problem specific features will be taken into account during the application of the ERX operator or refinement of the solution and they will replace the random selection.

The PSO algorithm itself can be made more elaborated, too. Random replacement of a particle can be introduced as another mechanism that brings stochastic element into the search. It allows the swarm to escape from the local optimum which can attract all particles.

Acknowledgments

The research has been done on the base of the research project GAČR-402/09/0405 funded by the Grant Agency of the Czech Republic. The support of the agency is kindly acknowledged.

References

[1] T. J. Ai and V. Kachitvichyanukul, A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery, *Computers &*

Operations Research, Vol. 36, No. 5, 2009, pp. 1693-1702.

[2] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, Wiley, 2006.

[3] M. Gendreau, G. Laporte, J.-Y. Potvin, Metaheuristics for the Capacitated VRP, In P. Toth and D. Vigo, (Eds.), *The Vehicle Routing Problem (Monographs on Discrete Mathematics & Applications)*, Society for Industrial & Applied Mathematics, 2001, pp. 129-154.

[4] G. Laporte and F. Semet, Classical Heuristics for the Capacitated VRP, In P. Toth and D. Vigo, (Eds.), *The Vehicle Routing Problem (Monographs on Discrete Mathematics & Applications),* Society for Industrial & Applied Mathematics, 2001, pp. 109-128.

[5] R. Poli, Analysis of the publications on the applications of particle swarm optimisation, *Journal of Artificial Evolution and Applications*, Vol. 2008, No. 1, 2008, pp 1–10.

[6] J.-Y. Potvin, A Review of Bio-inspired Algorithms for Vehicle Routing, In F. Baptista Pereira and J. Tavares (Eds.), *Bio-Inspired Algorithms for the Vehicle Routing Problem (Studies in Computational Intelligence),* Springer, 2008, pp. 1-34.

[7] P. Toth and D. Vigo, An Overview of Vehicle Routing Problems, In P. Toth and D. Vigo (Eds.), *The Vehicle Routing Problem (Monographs on Discrete Mathematics & Applications),* Society for Industrial & Applied Mathematics, 2001, pp. 1-26.

[8] D. Whitley, T. Starkweather, D. A. Fuquay, Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator, In J.D. Schaffer (ed.), *Proc. of the 3rd Intl. Conf. on Genetic Algorithms,* San Mateo, California, Morgan Kaufmann,1989, pp. 133-14