The 7th International Symposium on Operations Research and Its Applications (ISORA'08)
Lijiang, China, October 31–Novemver 3, 2008

Quadratic Programming for the Vehicle Routing Problem

Ping Ji Hongtao Wu Yongzhong Wu

Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China

Abstract The classical capacitated vehicle routing problem (VRP) is discussed in this paper. A quadratic programming formulation for the problem is proposed. Compared with a common mixed integer linear programming (MIP) formulation in the literature, the number of total constraints is significantly reduced through avoiding the subtour elimination constraints. The simplicity of the new model is beneficial for the formulation of more complicated VRP variants. It also introduces the possibility of using the well-known quadratic programming algorithms for solving the vehicle routing problem.

Keywords Vehicle routing problem; Quadratic programming

1 Introduction

The classical capacitated vehicle routing problem (VRP) is a hard combinatorial optimization problem, in which customers of known demand are supplied from a single depot with a fleet of V (> 1) vehicles of fixed loading capacity and/or with traveling time (distance) constraint. The problem consists of designing a set of at most V delivery or collection routes such that (1) each route starts and ends at the depot, (2) each customer is visited exactly once by exactly one vehicle, (3) the total demands of customers in each route does not exceed vehicle capacity, and the total time (distance) of each route does not exceed the allowed traveling time (distance) of each vehicle, and (4) the total routing cost (time or distance) is minimized.

The problem was introduced nearly 50 years ago by Dantzig and Ramser (1959)[2] and has since given rise to a rich body of research. Many real-life problems were found to be instances of the VRP, e.g., the delivery of newspapers to retailers, of food and beverages to grocery stores, and the collection of milk products from dairy farmers, of express mails from customers, etc.

The VRP is *NP*-hard because it includes a well-known *NP*-hard problem, the Traveling Salesman Problem (TSP) as a special case where there is only one vehicle without capacity or time (distance) limit. To this day, the VRP remains very difficult to solve optimally. The most successful exact algorithms for the VRP can only solve instances of up to about 100 customers, and with a variable success rate. As the result, most of the recent research effort has concentrated on heuristic methods. Laporte (2007)[3] conducted a comprehensive survey of the VRP solution methods in the literature.

For both exact algorithms and heuristic methods, an efficient formulation of the problem is much important for solving the real-life VRP instances, which have many additional complicated constraints. As mentioned by Marinakis et al. (2007)[4], an efficient formulation together with a powerful and efficient algorithm (usually metaheuristic) based on the properties of the model provides the opportunity to find a near-optimum solution without spending excessive computational time.

This paper focuses on developing a new quadratic programming model for the VRP, so that the well-established quadratic programming algorithms may be an alternative choice for solving the VRP. In addition, the new formulation is simplified through significant reduction in the number of constraints, and thus stands for greater chance of being used for the formulation of more complicated real-life problems.

2 A common MIP formulation

In the literature, the vehicle routing problem is usually formulated as a mixed integer programming (MIP) model with integer variables associated with each arc between locations (customers and the depot). This model is known as the Vehicle Flow Model (Bodin et al., 1983)[1]. The terminology used in the model is described as follows.

1. Parameters
   N: customers
   V: vehicles
   d_i: demand of customer i
   T_v: capacity of vehicle v
   c_{ij}: distance between node i and node j
2. Decision Variables
   x_{ij}^v : 1if vehicle v travels on arc $(i,\ j)$; otherwise 0
   $y_{ij:}$ 1 if a vehicle travels on arc $(i,\ j)$; otherwise 0
   $u_{i:}$ an intermediate variable, non-negative and real

With the above parameters and variables, a vehicle routing problem can be formulated with the objective of minimizing the total traveling distance of the vehicles:

$$z = \sum_{v=1}^{V} \sum_{i=0}^{N} \sum_{j=0}^{N} c_{ij} x_{ij}^v \tag{1}$$

The loading capacity of each vehicle cannot be exceeded and this is ensured by constraints (2):

$$\sum_{i=1}^{n} \sum_{j=0}^{n} d_i x_{ij}^v \leq T_v \ (v = 1, 2,\ \ldots,\ V) \tag{2}$$

Each arc $(i,\ j)$ can be traveled by at most one vehicle and this is ensured by constraints (3):

$$\sum_{v=1}^{V} x_{ij}^v = y_{ij} \ (i, j\ =\ 0,\ 1,\ \ldots,\ N) \tag{3}$$

Constraints (4) and (5) ensure that each customer must be visited exactly once:

$$\sum_{\substack{j=0 \\ j \neq i}}^{N} y_{ij} = 1 \; (i = 1, 2, \ldots, N) \tag{4}$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{N} y_{ij} = 1 \; (j = 1,2, \ldots, N) \tag{5}$$

Constraints (6) and (7) ensure that a vehicle must start from and end at the depot if it is in use, and at most V vehicles will be used.

$$\sum_{j=1}^{N} y_{0j} \leq V \tag{6}$$

$$\sum_{i=1}^{N} y_{i0} \leq V \tag{7}$$

Constraints (8) are the subtour elimination constraints to prevent the formation of subtours:

$$u_i - u_j + (N+1)y_{ij} \leq N \; (1 \leq i \neq j \leq N) \tag{8}$$

Constraints (9) ensure the vehicle continuity, i.e., a vehicle that reaches a customer must leave the same customer.

$$\sum_{\substack{j=0 \\ j \neq i}}^{N} x_{ij}^{v} = \sum_{\substack{j=0 \\ j \neq i}}^{N} x_{ji}^{v} \;\; (i = 1, 2, ..., N; v = 1, 2, ..., V) \tag{9}$$

A problem instance is generated to explain the above MIP model in more details. In this instance, there are 5 customers to be serviced by 3 vehicles. Table 1 shows the demands by each customer, and Table 2 shows the loading capacities of each vehicle. The distances between each pair of nodes are shown in Table 3.

Table 1: Customer demands

Customer	1	2	3	4	5
Demand	350	400	300	750	400

The instance was formulated as a MIP model as described above (see Appendix) and solved by an existing MIP solver. The solution of the problem is illustrated in Figure 1. As shown in the figure, customer 5 and 1 are serviced by vehicle B sequentially, and

Table 2: Vehicle capacities

Vehicle	A	B	C
Capacity	1000	1000	1500

Table 3: Distances between each pair of nodes

To From	0	1	2	3	4	5
0[a]	∞[b]	9	14	21	23	22
1	9	∞	5	12	22	21
2	14	5	∞	7	3	16
3	21	12	7	∞	10	21
4	23	22	3	10	∞	19
5	22	21	16	21	19	∞

[a] Node 0 represents the depot

[b] ∞ means There is no are between the corresponding nodes

customer 2, 4, and 3 are serviced by vehicle C sequentially. Vehicle A is not in use. The total traveling distance of all the vehicles is 100.

One of the main drawbacks of the MIP formulation is the large number of variables and constraints. The explicit generation of all constraints is normally tedious. In the above model formulated for the instance, there are totally 149 variables (144 binary variables and 5 continuous variables) and 89 linear constraints.

3 A new quadratic programming (QP) formulation

In order to simplify the VRP formulation, a quadratic programming (QP) model is proposed.

In the QP formulation, new 0-1 variables, $x_{ip}^{v}$are defined, with x_{ip}^{v}= 1 representing that vehicle v visits customer i at step p and x_{ip}^{v}= 0 otherwise. With the new variables, the objective of the problem can be formulated as:

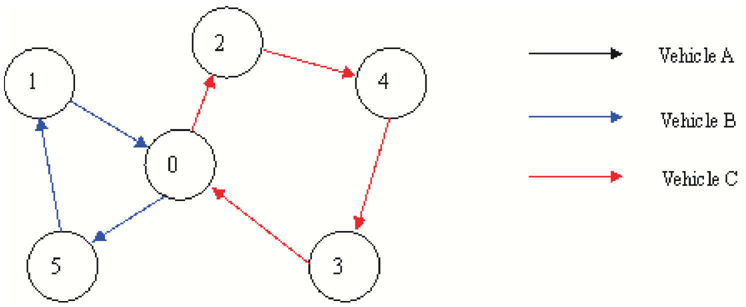

Figure 1: An example of the VRP

$$\text{MIN}\, z = \sum_{v=1}^{V}\sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{p=1}^{N} c_{ij}x_{ip}^{v}x_{ip+1}^{v} \tag{10}$$

A customer must be visited exactly once by exactly one vehicle and this is ensured by constraints (11):

$$\sum_{v=1}^{V}\sum_{p=1}^{N} x_{ip}^{v} = 1 \quad (i = 1, 2, \cdots, N) \tag{11}$$

A vehicle reaching a customer must leave for another customer or return to the depot and this is ensured by constraints (12):

$$\sum_{i=1}^{N} x_{ip}^{v} = \sum_{i=0}^{N} x_{ip+1}^{v} \quad (p = 1, 2, \cdots, N; v = 1, 2, \cdots, V) \tag{12}$$

The total demands a vehicle delivers cannot exceed its capacity and this is ensured by constraints (13):

$$\sum_{i=1}^{N} d_i \sum_{p=1}^{N} x_{ip}^{v} \le T_v \quad (v = 1, 2, \cdots, V) \tag{13}$$

In order to compare the new QP model with the MIP model, the same problem instance described above was tested. A quadratic programming solver was used to solve the QP formulation of the problem instance. The same solution with that for the MIP model was obtained.

In the QP model of the instance, there are totally 111 binary variables and 23 linear constraints. The formulation complexity of the new QP model is reduced comparing to the MIP model in that the number of constraints is greatly reduced from 89 to 23. This is mainly because that the new variables x_{ip}^{v} implicitly prevent the formation of subtours, so the complicated subtour elimination constraints (8) in the MIP model are not required by the new QP model. On the other hand, the computational effort required for solving the problem is not reduced because it introduces new quadratic terms in the objective function.

The significance of the new QP model is twofold:

1. It introduces the possibility for using the well-known QP methods for solving the vehicle routing problem.

2. It can be used for efficient formulation of more complicated VRP variants with additional constraints, e.g., the time window constraints.

4 Summary

A new quadratic programming (QP) formulation is proposed for the capacitated vehicle routing model. Compared to a common MIP formulation in the literature, the new QP formulation has greater simplicity through avoiding the complicated subtour elimination constraints. The simplicity is beneficial for efficient formulation of more complicated VRP variants. In addition, with the new formulation, the QP solution methods may be useful for solving the vehicle routing problems.

Acknowledgement

The authors acknowledge The Hong Kong Polytechnic University for the financial support of the project (G-U353).

References

[1] Bodin, L., Golden, B., Assad, A., and Ball, M., 1983. The state of the art in the routing and scheduling of vehicles and crews. Computers and Operations Research 10: 63–212.

[2] Dantzig G.B and Ramser J.H., 1959. The truck dispatching problem, Management Science 6: 80–91.

[3] Laporte G., 2007. What you should know about the vehicle routing problem. Naval Research Logistics 54: 811-819.

[4] Marinakis Y., Migdalas A., and Pardalos P. M., 2007. A new bilevel formulation for the vehicle routing problem and a solution method using a genetic algorithm. Journal of Global Optimization 38:555-580.

5 Appendix

MIN z = 99999x00A + 9x01A + 14x02A + 21x03A + 23x04A + 22x05A
+ 9x10A + 99999x11A + 5x12A + 12x13A + 22x14A + 21x15A
+ 14x20A + 5x21A + 99999x22A + 7x23A + 3x24A + 16x25A
+ 21x30A + 12x31A + 7x32A + 99999x33A + 10x34A + 21x35A
+ 23x40A + 22x41A + 3x42A + 10x43A + 99999x44A + 19x45A
+ 22x50A + 21x51A + 16x52A + 21x53A + 19x54A + 99999x55A
+ 99999x00B + 9x01B + 14x02B + 21x03B + 23x04B + 22x05B
+ 9x10B + 99999x11B + 5x12B + 12x13B + 22x14B + 21x15B
+ 14x20B + 5x21B + 99999x22B + 7x23B + 3x24B + 16x25B
+ 21x30B + 12x31B + 7x32B + 99999x33B + 10x34B + 21x35B
+ 23x40B + 22x41B + 3x42B + 10x43B + 99999x44B + 19x45B
+ 22x50B + 21x51B + 16x52B + 21x53B + 19x54B + 99999x55B
+ 99999x00C + 9x01C + 14x02C + 21x03C + 23x04C + 22x05C
+ 9x10C + 99999x11C + 5x12C + 12x13C + 22x14C + 21x15C
+ 14x20C + 5x21C + 99999x22C + 7x23C + 3x24C + 16x25C
+ 21x30C + 12x31C + 7x32C + 99999x33C + 10x34C + 21x35C
+ 23x40C + 22x41C + 3x42C + 10x43C + 99999x44C + 19x45C
+ 22x50C + 21x51C + 16x52C + 21x53C + 19x54C + 99999x55C

SUBJECT TO

x00A + x00B + x00C – y00 = 0
x01A + x01B + x01C – y01 = 0
x02A + x02B + x02C – y02 = 0
x03A + x03B + x03C – y03 = 0
x04A + x04B + x00C – y04 = 0
x05A + x05B + x05C – y05 = 0
x10A + x10B + x10C – y10 = 0

x11A + x11B + x11C – y11 = 0
x12A + x12B + x12C – y12 = 0
x13A + x13B + x13C – y13 = 0
x14A + x14B + x14C – y14 = 0
x15A + x15B + x15C – y15 = 0
x20A + x20B + x20C – y20 = 0
x21A + x21B + x21C – y21 = 0
x22A + x22B + x22C – y22 = 0
x23A + x23B + x23C – y23 = 0
x24A + x24B + x24C – y24 = 0
x25A + x25B + x25C – y25 = 0
x30A + x30B + x30C – y30 = 0
x31A + x31B + x31C – y31 = 0
x32A + x32B + x32C – y32 = 0
x33A + x33B + x33C – y33 = 0
x34A + x34B + x34C – y34 = 0
x35A + x35B + x35C – y35 = 0
x40A + x40B + x40C – y40 = 0
x41A + x41B + x41C – y41 = 0
x42A + x42B + x42C – y42 = 0
x43A + x43B + x43C – y43 = 0
x44A + x44B + x44C – y44 = 0
x45A + x45B + x45C – y45 = 0
x50A + x50B + x50C – y50 = 0
x51A + x51B + x51C – y51 = 0
x52A + x52B + x52C – y52 = 0
x53A + x53B + x53C – y53 = 0
x54A + x54B + x54C – y54 = 0
x55A + x55B + x55C – y55 = 0
y01 + y11 + y21 + y31 + y41 + y51 = 1
y02 + y12 + y22 + y32 + y42 + y52 = 1
y03 + y13 + y23 + y33 + y43 + y53 = 1
y04 + y14 + y24 + y34 + y44 + y54 = 1
y05 + y15 + y25 + y35 + y45 + y55 = 1
y10 + y11 + y12 + y13 + y14 + y15 = 1
y20 + y21 + y22 + y23 + y24 + y25 = 1
y30 + y31 + y32 + y33 + y34 + y35 = 1
y40 + y41 + y42 + y43 + y44 + y45 = 1
y50 + y51 + y52 + y53 + y54 + y55 = 1
y00 + y10 + y20 + y30 + y40 + y50 <= 3
y00 + y01 + y02 + y03 + y04 + y05 <= 3

350x10A + 350x11A + 350x12A + 350x13A + 350x14A + 350x15A
+ 400x20A + 400x21A + 400x22A + 400x23A + 400x24A + 400x25A

+ 300x30A + 300x31A + 300x32A + 300x33A + 300x34A + 300x35A
+ 750x40A + 750x41A + 750x42A + 750x43A + 750x44A + 750x45A
+ 400x50A + 400x51A + 400x52A + 400x53A + 400x54A + 400x55A <= 1000
350x10B + 350x11B + 350x12B + 350x13B + 350x14B + 350x15B
+ 400x20B + 400x21B + 400x22B + 400x23B + 400x24B + 400x25B
+ 300x30B + 300x31B + 300x32B + 300x33B + 300x34B + 300x35B
+ 750x40B + 750x41B + 750x42B + 750x43B + 750x44B + 750x45B
+ 400x50B + 400x51B + 400x52B + 400x53B + 400x54B + 400x55B <= 1000
350x10C + 350x11C + 350x12C + 350x13C + 350x14C + 350x15C
+ 400x20C + 400x21C + 400x22C + 400x23C + 400x24C + 400x25C
+ 300x30C + 300x31C + 300x32C + 300x33C + 300x34C + 300x35C
+ 750x40C + 750x41C + 750x42C + 750x43C + 750x44C + 750x45C
+ 400x50C + 400x51C + 400x52C + 400x53C + 400x54C + 400x55C <= 1500
x01A + x02A + x03A + x04A + x05A – x10A – x20A – x30A – x40A – x50A = 0
x10A + x12A + x13A + x14A + x15A – x01A – x21A – x31A – x41A – x51A = 0
x20A + x21A + x23A + x24A + x25A – x02A – x12A – x32A – x42A – x52A = 0
x30A + x31A + x32A + x34A + x35A – x03A – x13A – x23A – x43A – x53A = 0
x40A + x41A + x42A + x43A + x45A – x04A – x14A – x24A – x34A – x54A = 0
x50A + x51A + x52A + x53A + x54A – x05A – x15A – x25A – x35A – x45A = 0
x01B + x02B + x03B + x04B + x05B – x10B – x20B – x30B – x40B – x50B = 0
x10B + x12B + x13B + x14B + x15B – x01B – x21B – x31B – x41B – x51B = 0
x20B + x21B + x23B + x24B + x25B – x02B – x12B – x32B – x42B – x52B = 0
x30B + x31B + x32B + x34B + x35B – x03B – x13B – x23B – x43B – x53B = 0
x40B + x41B + x42B + x43B + x45B – x04B – x14B – x24B – x34B – x54B = 0
x50B + x51B + x52B + x53B + x54B – x05B – x15B – x25B – x35B – x45B = 0
x01C + x02C + x03C + x04C + x05C – x10C – x20C – x30C – x40C – x50C = 0
x10C + x12C + x13C + x14C + x15C – x01C – x21C – x31C – x41C – x51C = 0
x20C + x21C + x23C + x24C + x25C – x02C – x12C – x32C – x42C – x52C = 0
x30C + x31C + x32C + x34C + x35C – x03C – x13C – x23C – x43C – x53C = 0
x40C + x41C + x42C + x43C + x45C – x04C – x14C – x24C – x34C – x54C = 0
x50C + x51C + x52C + x53C + x54C – x05C – x15C – x25C – x35C – x45C = 0
u1 – u2 + 6y12 <= 5
u1 – u3 + 6y13 <= 5
u1 – u4 + 6y14 <= 5
u1 – u5 + 6y15 <= 5
u2 – u1 + 6y21 <= 5
u2 – u3 + 6y23 <= 5
u2 – u4 + 6y24 <= 5
u2 – u5 + 6y25 <= 5
u3 – u1 + 6y31 <= 5
u3 – u2 + 6y32 <= 5
u3 – u4 + 6y34 <= 5
u3 – u5 + 6y35 <= 5

u4 – u1 + 6y41 <= 5
u4 – u2 + 6y42 <= 5
u4 – u3 + 6y43 <= 5
u4 – u5 + 6y45 <= 5
u5 – u1 + 6y51 <= 5
u5 – u2 + 6y52 <= 5
u5 – u3 + 6y53 <= 5
u5 – u4 + 6y54 <= 5

All *x*'s and *y*'s variables are binary while all *u*'s are real.