# Solving Vehicle Routing Problem with Earliness and Lateness Costs by Linear Programming Cooperated with Constraint Programming

03000500 アイログ(株) CHENG, Yu

## 1. Introduction

There are many real-world, industrial problems where the possible solutions are so numerous that it is not practical to consider all of them in a search for an optimal feasible solution. These problems are sometimes referred to as combinatorial optimization problems, emphasizing the idea that the number of possible solutions grows combinatorially as the number of decision variables increases. There are many different approaches to solve such kinds of optimization problems. Linear programming and its extension to integer programming are one of the most well-known approaches. Constraint programming is a relatively new technique that has been proved particularly powerful when used to solve combinatorial problems with complicated constraints. Both of them are widely used throughout industry and operations research.
A new approach and practical way of attacking such intractable problems is through constraint programming techniques to reduce the combinatorial explosion and at the same time use integer programming techniques for any linear constraints in the model to speed the search of solution. Thus, the approach suggested and introduced by this paper is to combine constraint programming with linear programming to solve complex combinatorial optimization problems. By this approach, both constraint programming and integer programming make use of the structure of a given combinatorial optimization problem, and hybrid cooperating optimizers are used to provide facilities to combine these two techniques in order to take advantage of both.

## 2. Linear programming

Linear programming (LP) is probably the best known and most widely used optimization technique. Any problem that can be formulated with real decision variables, a linear objective function, and linear constraint functions, can be solved using linear programming. This approach has been tremendously successful in solving a broad range of commercial resource allocation problems.
Integer linear programming and mixed integer linear programming use linear programming to solve problems with some integer variables (all other variable types may be represented by a combination of integer variables and linear constraints), but still having linear objective and constraint functions. The integer variables are represented as real variables, and the resulting linear program is solved; then a repetitive process is used in which an integer variable is bounded above or below in an attempt to force it to an integer value by adding constraints, and the modified linear program is resolved. This method (branch and bound) terminates when all the integer variables take integer values. When the number of integer variables is small, integer programming solves problems fairly quickly; unfortunately, this process can be too time consuming for problems with large numbers of integer variables. Some problems require millions of iterations to be solved.

## 3. Constraint Programming

Constraint programming (CP) is a way of solving constraint satisfaction problems, which consists of a number of variables and a number of relations on and among those variables. A constraint is a mathematical relation between possible values of variables. The set of possible assignments of values to variables is known as the search space. Finding solution to a CP modeled problem is to assign values to the constraint variables of the problem in such a way that all the constraints imposed on the variables are satisfied simultaneously. CP is efficient because, rather than searching that search space blindly, it exploits the constraints themselves by constraint propagation to reduce its effort in the search. Thus, the constraint propagation is known as the most powerful features of the CP. Another advantage of CP in attacking combinatorial optimization problems is that CP offers a variety of modeling facilities, such as logical constraints, higher-order constraints, and global constraints, which make it somewhat easier to formulate a model that is natural and intuitive for people. These natural, intuitive models make it easier to exploit the problem structure, for example, in global or logical constraints, to find a solution more efficiently. Each time a new search decision is made, all the relevant constraints (linear and nonlinear) are automatically propagated in order to reduce further exploration of the search space. CP also facilitates the design of search procedures that exploit the structure of a given problem. For example, user's search goals can be formulated to guide a search, and some predefined evaluators and selectors can also be used for controlling the search effectively.
However, the weakness of this approach becomes apparent when searching the feasible region for an optimal solution. If the feasible region is large, this approach may become inefficient at finding the best solution.

## 4. Hybrid Cooperating Optimizers --Cooperating LP with CP

LP and CP are proved to be successful in solving complementary classes of problems; LP when optimization is the major aspect, and CP when feasibility is the critical concern. In LP, all the constraints are evaluated at the same time, while CP evaluates the effect of constraints sequentially and propagation it through variable domains. Due to this, generally, it is difficult to get optimal solution by CP for loosely constrained problems. On the other hand, LP requires all the constraints to be linear, while this restriction does not apply for CP, in which various kinds of constraints are allowed and it is very good for modeling complex problems. Obviously, the combination of CP with LP to form a hybrid cooperating optimizer would be a better approach when solving some complex problems with both real and integer variables and linear and other types of constraints. This approach may take advantages from both of them and speeds up the search for the solutions.

There are several different approaches for using CP and LP together. Some approaches use the two simultaneously; others apply the them sequentially (one after the other). Normally, with a hybrid cooperating optimizer it is possible to maintain a (optimal) relaxed solution automatically to linear constraints while taking into account the modification of variables--either manually or when CP's propagation engine deduces new bounds for them while processing other constraints. By this way, it provides an encapsulation of LP in a global constraint, which helps to guide the search, to tighten variable bounds for better constraint propagation, and to achieve earlier detection of infeasibilities. With a hybrid cooperating optimizer, a problem can contain a mix of logical and linear constraints, and a search procedure developed with CP can treat the set of linear constraints as a single constraint. Finally, the domain reduction and constraint propagation procedures cause communication to automatically occur between the different kinds of constraints. Thus, hybrid cooperating optimizers that combine the power of CP and LP and try to use their complementary strengths in an efficient way may be able to solve problems that would be unsolvable using just a single methodology.

## 5. Case study

A vehicle routing problem with earliness and lateness costs is used here as a case study to show the effect of cooperating LP with CP.

The problem considers deliveries to customers within time windows that are not hard but may be violated at a cost. In real life situations, a late delivery may cost the customer as a lack of stock leads to lost sales; an early delivery may increase a customer's storage and handling costs due to excessive stock levels. Because the Vehicle Routing Problem (VRP) here deals with costs for both early and late deliveries, its difficulty is greater than that of a VRP with lateness costs only, where the optimal arrival time is simply the earliest possible. In this problem, the visits in a route are more interdependent. Making an initial delivery on time can cause subsequent deliveries to be late and costly; making the subsequent deliveries on time can cause the initial delivery to be costly as it is too early.

For a vehicle routing problem, the most natural approach is to solve it by using a two-phase approach based on the search for a first solution and the improvement of this solution by using local search algorithms, such as tabu search etc. These phases can themselves be decomposed into two steps: first, the next-variables are instantiated to create sequences of visits that form routes; second, the arrival times at these visits are set. This second step is crucial in the local search phase to determine if there exist arrival times for the current routes that improve the cost.

However, this approach has a severe drawback. Local search can only improve the solution by very small steps, as no attempt is made to minimize the global cost of arrival times, completely stalling the search process. This could be dealt with using a variable selector, but the resulting search tree would be impractically huge.

The preferred alternative is to use LP to solve arrival time related cost sub-problem optimally, as the model is completely linear once the routes have been built. Thus, using hybrid to combine the capabilities of CP and LP is a good way to approach the resolution of these problems. First, CP can be used to generate routes with little concern for costs arising from early or late deliveries, and then LP can be used to optimize the earliness and lateness costs of the routing problem.

A comparison of two approaches is shown in table 1. With the condition of the limited number of iterations on improvement of solutions, which equals to 30, it shows that when using both CP and LP, the best cost was found very quickly than when using CP only. It can be seen that with the help information from LP, the memory used is smaller, and the number of choice points created for the search has been reduced greatly.

| Evaluations items | CP only | LP and CP |
|---|---|---|
| Best cost found | 3020.26 | 430.848 |
| Number of used vehicles | 3 | 3 |
| Number of choice points | 1073 | 6 |
| Memory used (bytes) | 517816 | 308776 |

Table 1: A comparison of two approaches.

In this case study, ILOG Dispatcher, a CP-based routing generator, is used to build routes, and ILOG CPLEX, a LP solver, is used to schedule optimal arrival times, thus it forms a hybrid cooperating optimizer and the problem is handled and solved using a combination of CP and LP. This hybrid cooperating optimizer can be used to build industrial applications for production planning, resource allocation, staffing, blending, and a host of other problems.

## 6.Remarks

In the paper, a way of linear programming algorithm cooperated with constraint programming is suggested to solve combinatorial optimizations more quickly and more efficiently. As a result, combining these two techniques provides a more efficient approach that may be applied to a broad range of application problems.

## 7. Reference


[1] ILOG Optimization Technology White Paper, ILOG Inc. 2001. (http://www.ilog.co.jp/)

[2] ILOG Solver, User Manual, version 5.2, 2002.

[3] ILOG CPLEX, User Manual, version 8.0, 2002.

[4] ILOG Dispatcher, User Manual, version 3.2, 2002.

[5] ILOG Hybrid Cooperating Optimizers, User Manual, version 1.3, 2002.