6th International Conference on Industrial Engineering and Industrial Management.
XVI Congreso de Ingeniería de Organización. Vigo, July 18-20, 2012

A Multi-start algorithm for solving Ready Mix Concrete Production and Delivery Scheduling Problem (RMCPDSP)

Bellés A, Coves AM [1] , de los Santos MA

Abstract: In this paper we present a MS algorithm for solving RMCPDSP. Integration of production and delivery is critical in the case of Ready Mix Concrete, as it expires approximately an hour and a half after it is produced. This MS algorithm is based on specifically designed coding that, upon simulation, provides the goodness-of-fit of the analysed solution. Examples have been generated on the basis of real case data. The obtained results validate the efficiency of the MS algorithm.

Key words: production and delivery scheduling, ready mix concrete**,** Multi-start algorithm

1 Introduction

Research interest in RMCPDSP rose in 2004 (Matsatsinis, 2004) with the upsurge in economic activity in the building sector, which forced construction companies to improve their efficiency and lower costs. During the economic recession beginning in 2008, a number of publications dealing with RMCPDSP appeared. The most noteworthy were two studies: one that proposed a genetic algorithm for solving RMCPDSP (Naso et al., 2007) and the other which analysed the problem in depth, modelling it with MILP (Mixed Integer Linear Programming) and proved RMCPDSP to be NP-hard (Asbach et al., 2009). Research shows that ready mix concrete manufacturing and delivery companies must optimize production and delivery resource scheduling in order to increase efficiency and lower costs.

[1] Anna M. Coves (✉)
Department of Management, Institute of Industrial and Control Engineering, Universitat Politècnica de Catalunya, Av. Diagonal 647, 08028 Barcelona, Spain
e-mail: anna.maria.coves@upc.edu

The ready mixed concrete—RMC-- consists of coarse and fine aggregates, cement, sand, water and specified admixtures. It is used for construction, such as pillars, beams, slabs and walls, etc. The fresh RMC has a life span of 90 minutes and thus cannot be produced and stored in advance. It is a perishable and non-storable material.

The production process is carried out in the plants, where the cement, aggregates, water, and additives are introduced into the mixing plant hoppers in order to obtain concrete. Once loaded on the trucks and in transit, the RMC is kept in constant rotation.

In most cases, the precise quantity, quality and type of product and service requested is unknown until only shortly before the delivery date—thus greatly complicating the scheduling process. Every day deliveries for the following day are scheduled.

In this paper we propose a MS algorithm for scheduling production and orders delivery for the following day.

This paper first describes the problem, and then goes on to describe the solution coding. Solution coding will be the base of the proposed MS algorithm. Examples based on data from a real case will be generated, which as will be demonstrated in the pages ahead, validate the efficiency of the MS algorithm. In the conclusion, future lines of research are proposed.

2 Daily Production and Delivery Scheduling Problem

This section examines the characteristics of the RMCPDSP. These characteristics must be considered when designing algorithms to optimize the processes cited in the earlier section of this paper.

The critical resource in production is plant capacity, which is here understood as the number of load lines and the time spent loading a vehicle. The critical resource in delivery is the number of available vehicles. Production and delivery scheduling consists of assigning orders to productive plants; prioritizing these orders according to certain pre-established criteria; and finally, determining a starting and ending time for each operation, be it for a plant or for a vehicle. Effective scheduling requires: knowing the capacity of each plant; the distance of potential supplier plants from customers; the volume of each order; the number and frequency of trips necessary to fill orders; as well as specific time windows needed at the time of initial delivery of each order.

At the end of day, the company must schedule the resources for the following day, including: plant assignment; 2) number of vehicles needed in each plant; and 3) both load lines and vehicles scheduling.

To solve this problem, we design the MS algorithm based on a proposed solution code described in the following section.

3 RMCPDSP Solution Coding

This section describes the design of the solution code. After applying the proposed routine (hereafter referred to as the SIM routine) to the solution code, described in chapter 3.2, a complete solution will be obtained together with values determining the solution's good ness-of-fit

The solution code design process begins by determining the variables to be considered to determine the complete solution related to the code. To illustrate the code, an example will be used that is on a small-scale, but shares the characteristics and hypotheses of the problem under study.

In the presented example, there are 3 production plants that must produce and deliver 10 orders. Each plant has one load line, except plant 3 which has 2 load lines. The loading time is 10 minutes in plants 1 and 3, and 6 minutes in plant 2. Table 3.1 shows information regarding each of the orders. This information corresponds to the total volume requested per order. On the basis of this information, and on the load capacity of the trucks utilized, the number of trips required to fill the order is determined. The time of the first delivery of each order is known, as is the accepted delivery time windows and trip frequency.

Table 3.1 Order data

Order	Volume (m^3)	Trips	Time of the first delivery	Windows time (min)	Trip frequency (min)
1	20	3	09:54	18	13
2	19	3	09:50	16	15
3	17	3	15:39	17	11
…	…	…	…	…	…

3.1 Solution Code

The solution code contains the variables of the problem under study. In this paper, three sets of variables have been considered in which each set indicates one of the following concepts: *plant* order assignment; delivery *priority*; and *time* of first delivery for each order. This last concept will be subsumed within the accepted time window for each order.

Table 3.2 shows a code for the presented example with values given for the three sets of variables corresponding with the lines: plant, priority and time, respectively. As can be observed, order 1 has been assigned to plant 2, with a priority of 8, and a delivery time of 9:57 am.

Table 3.2 Example solution code

Order	*Plant*	*Priority*	*Time of the first delivery*
1	2	8	09:57
2	2	1	09:35
3	3	6	15:47
…	…	…	…

3.2 Complete Solution (SIM Routine)

Using the solution code, a simulation running the SIM routine determines the complete solution.

The complete solution integrates plant and delivery scheduling. Thus the total solution includes truck assignment for each order, scheduling of trips, including initial loading time at the plant and return time to plant after delivery. The complete solution also contains information on the number of load lines needed in each plant to carry out the delivery. If the number of load lines needed is no greater than those available, the solution is feasible.

The complete solution is obtained applying the SIM routine, which assigns the trucks for each order and plant. Truck assignments may be reassigned to other plants throughout the process, depending on delivery departure times.

The hypothesis is that the available fleet of vehicles and number of plant load lines is infinite. The SIM routine initiates the assignment of trucks for maximum priority orders. Whenever possible, trucks already assigned are used, even if that implies changing plants. When this is not possible, a new truck is assigned. Availability of trucks that have already been assigned for a trip is determined on the basis of total travel time of previously assigned trips. The number of different vehicles assigned to orders determines the size of the fleet necessary to deliver orders. The number of load lines needed in each plant is also determined. In Fig. 3.1 there is a SIM routine pseudo code.

The complete solution for solution code of the example (table 3.2) has been obtained by applying SIM routine. Values for delivery distances have been generated on the basis of real case data of a company in the sector. The number of extra load lines that each of the plants must have in order to deliver orders is one for plant 2. For this example solution code, the complete solution is not feasible.

```
Step 1: Initialize data for orders, plants, and trucks.
Step 2: Initialize solution code.
Step 3: Orders are prioritized.
Step 4: For (each order)
            For (each trip for each order) set time for starting and ending trip
                    If (Only one truck is available for trip)
                            A trip is assigned to a truck
                    else if (More than one truck is available for the trip)
                            The first available truck is assigned.
                    else if (No truck is available in any plant for the trip).
                            A new truck is "created" and a trip is assigned to it.
                    End if
            End For
        End For
Step 5: For (each trip)
            - Truck arrival time at the plant is set for those trucks that-- instead of re-
            turning to plant of origin--go to another plant to load up for another deliv-
            ery (if there is one).
        End For
Step 6: For (each plant)
            - Determine if more load lines than those existing at the plants are needed,
            and if so, how many.
```

Fig.3.1 *SIM routine* pseudo code

3.3 Objective Function

To assess a complete solution, a hierarchical objective function has been established. The expression of the objective function is the following:

$$[\text{Min}]\ Z = 100000\ a + 1000\ b + 20\ c + 1{,}5\ e \tag{3.1}$$

Where:

a : total number of load lines needed beyond those available.

b: total number of vehicles to deliver orders.

c: total time plants are open.

d: travel time of all trucks used

c: total truck travel distance.

A very high coefficient has been used for the variable, *a,* in order to reject infeasible solutions. Variable *b* has the second highest value of the objective function in order to give priority to solutions that require fewer vehicles to deliver orders.

The following values have been obtained for the example a=1; b=9; c=20,23; d=26,3; e=696. Thus the objective function value for this solution (infeasible) is 110.711,67.

4 Designed MS Algorithm

In order to obtain a good solution for RMCPDS problem, we have designed a heuristic algorithm based on Multi- start (MS algorithm) (Glover et al., 2003). MS algorithm is based on generating new solutions and assessing them in the manner presented in the previous section.

Variable values and solution coding are randomly generated on the basis of real case data. Given a set number of plants and orders, orders are randomly assigned to plants and randomly prioritized. Scheduled time for first delivery is randomly generated within an interval determined by the delivery time and time windows.

Once the solution code is generated, a complete solution is determined using SIM routine. The initial solution is considered for the moment as the best solution. Other solutions are then generated, which are kept or rejected depending on if they are better or not than the best solution. The final solution will be the best of those found within the time limits for running the algorithm. The MS algorithm pseudo code is the following in Fig. 4.1.

```
Step 1:  Initialize data for orders, plants, and truck.
Step 2: While (time < maximum running time)
            - Random assignment of values determining solution code
            - SIM routine application for obtaining complete solution.
            - Assessment of objective function of complete solution.
            If (current solution is better than best solution found thus far)
                    Current solution becomes the best solution found thus far.
            End if
          End While
```

Fig. 4.1 *MS* pseudo code**.**

The MS algorithm has been applied to the example in this paper for a maximum running time of 5 minutes. The solution described in the earlier section has been taken as the initial solution. The best solution obtained. The solution obtained after applying the MS algorithm is a feasible solution. The number of lines needed to deliver orders matches those existing at the plants. The values for calculating this solution's objective function are: a=0; b=6; c=16,33; d=23,57; e=548;

such that the objective function value of the best solution obtained with the MS algorithm is 7.384,33.

5 Computational Experiment and Results

To validate the algorithms, data from different work days, at different levels of production, have been collected. The level of production at a plant depends directly on demand (i.e. on the number and volume of orders). Two computational experiments were carried out. In the first experiment A) 30 randomly generated cases corresponding to 30 different days were solved. In the second experiment B) we have chosen 10 test cases corresponding to 10 days described and solved previously by another author (Fenoy, 2008). Both experiments were solved with the MS algorithm.

5.1 Computational Experiment A

30 cases have been randomly generated corresponding to 30 days of high demand. The parameters used in the 30 cases are described in table 5.1 and uniformly distributed random values are generated within the corresponding windows.

Table 5.1 Parameters and windows

Parameter	*Windows*
Number of Plants	3- 5
Volume (m^3)	15- 25
Windows time (min)	15- 20
Frequency among travels of the same order(min)	10 - 15
Number of lines per plant	1 - 2
Plant load time (min)	5 - 10

The number of orders has also been randomly generated using: the number of all the load lines and the average plant loading times. To generate the time for the first delivery of each order, a delivery schedule based on real delivery schedule data provided by the company has been used.

We have observed the evolution of the best solution (the average of the 30 cases) during the 5-minute search and we conclude that, MS algorithm satisfactorily solves a real case of RMCPDS problem.

5.2 Computational Experiment B

Instances of experiment B have been chosen from 10 cases described and solved previously using a hierarchical method based on Constraint Logical Programming (CLP) (Fenoy, 2008). The hierarchical CLP is used in order to minimize the number of trucks required for deliveries. Using our MS algorithm we are able to reduce the number of trucks required by 35%.

6 Conclusions

The solution coding proposed in this article allows for an agile and rapid random generation of solutions, and using the SIM routine, the attainment of a complete solution and its objective function value.

The experiment validates the efficacy of the MS algorithm in both A and B computational experiments. The MS algorithm effectively solves the RMCPDS, and provides good solutions in a maximum of 10 minutes for real cases.

Future lines of research are to continue exploring new metaheuristics. These include algorithms based on VNS; mixed algorithm applying MS and VNS sequentially; and hybrid MS+VNS algorithms. It is also desirable to advance in incorporating more characteristics than currently exist in real cases and proposing a RMCPDSP classification.

7 References

Asbach L, Domdof U, Pesh E. (2009) Analysis, modeling and solution of the concrete delivery problem. European Journal of Operational Research v193 pp 820–835

Fenoy I (2008). Modelo de Programación de Restricciones para la optimización de la distribución de una empresa del sector de la construcción. Academic work document. http://upcommons.upc.edu/pfc/handle/2099.1/5711

Glover F, Kochenberger G. (2003) Handbook of Metaheuristics. section 12: Martí, R. Multi-Start Methods

Matsatsinis NF. (2004). Towards a decision support system for the ready concrete distribution system: A case of a Greek company. European Journal of Operational Research 152, 487-499.

Naso D, Surico M, Turchiano B, Kaymak U. (2007) Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete. European Journal of Operational Research v177, 2069-2099.