A Constraint Programming Approach to Optimise Sugarcane Rail Operations

Mahmoud Masoud, Erhan Kozan and Geoff Kent

Queensland University of Technology
GPO Box 2434, Brisbane Qld 4000, Australia
m.masoud@student.qut.edu.au, e.kozan@qut.edu.au, g.kent@qut.edu.au

Abstract: *In Australia, railway systems play a vital role in transporting the sugarcane crop between farms and mills. The sugarcane transport system is very complex and uses a daily schedule, consisting of a set of locomotives runs, to satisfy the requirements of the mill and harvesters. In this paper, a more efficient sugarcane transport scheduling system is proposed to reduce the cost of sugarcane transport. A job shop scheduling approach is used to formulate this complicated system and constraint programming is applied to obtain feasibility of alternative, to deal with conflicting objectives and to produce global optimal solutions. Optimization Programming Language (OPL) is used to search for possible solutions to problems and for the best (optimal) solution based on specific criteria. A numerical investigation is presented and demonstrates that high quality solutions are obtainable for industry-scale applications.*

Keywords**:** *job shop scheduling, constraint programming, sugarcane railway.*

1. INTRODUCTION

The sugarcane transport system is an important element in the raw sugar production system. Most sugarcane crops in Australia are transported from farms to mills by railway. Sugarcane is transported in specially designed wagons called cane bins. The railway system can generally operate for 24 hours a day, while the harvesting period is limited to about 12 hours in the day. A cane railway network uses a single track and includes many branch lines or segments and a lot of sidings. The cane railway system performs two main tasks. Firstly, delivering empty bins from the mill to the harvesters at sidings. Secondly, collecting the full bins of cane from harvesters and transporting them to the mill. From the perspective of the transport system, the mill serves the function of converting full bins into empty bins, while the harvesters convert empty bins to full bins.

Sidings have a finite capacity where the empties and fulls are delivered and collected that cannot be exceeded. Each siding has a daily allotment of bins that often exceeds the capacity of the siding where the total delivers, or total collections, have to equal the siding allotment. Each locomotive can haul a limited number of empty bins and full bins; this number depends on the capacity of each locomotive. For safety conditions, the locomotive is not allowed to have a mix of empties and fulls. As a result, all empties must take place before fulls.

The transport sector has a big impact on the overall costs of a sugarcane production system. The total cost of the sugarcane transport operations is very high; over 35% of the total cost of sugarcane production in Australia is incurred in cane transport.

Masoud et al. 2010 developed a new model to solve the sugarcane rail transport system problem. In this model, binary integer programming was used to formulate the problem. The blocking sections were considered in this model to prevent any conflicts through the single railway. In real life, many branches (segments) of the railway do not have passing loops in order to ease passing locos. For that reason in the current paper, the aim of the blocking segments is to remove any conjunction on the railway.

This study will consider most of the sugarcane transport system constraints and develop new models to produce flexible and realistic schedules. Job shop scheduling (JSS) approach and constraint programming (CP) technique are used for solving complex combinatorial optimization problems. The integration of these approaches can produce optimal solutions in a reasonable time. Our problem will be formulated by both of these techniques and using ILOG OPL software to obtain results.

2. Constraint Programming (CP) Formulation for The sugarcane Transport System

In this paper, CP search techniques are used to obtain optimal solutions for research problems. CP has many advantages such as: CP has the ability to model many different types of combinatorial optimisation problems while mathematical programming techniques individually have restrictive modelling power. It can produce global optimal solutions. Many problems in real life need multiple solutions; CP techniques have the capabilities to generate many solutions. CP techniques have the ability to solve all the feasibility problems and can deal with conflicting objectives.

Modelling any problem using CP depends on the CP package used because of the differences in constructs available in various modelling languages. In this study, ILOG's OPL modelling language will be used. OPL has a set of constructs to scheduling problems to develop an effect CP model on the scheduling problems. The basic OPL modelling framework includes a set of commands that can model the elements of scheduling problems. The segments and sections are the resources, and they are modelled as *unary resources* in OPL. Each run is a unary resource because it is assigned for one locomotive as well. The characteristics of the unary resources are that they can process only one operation at the same time. Each operation is associated with a start time and duration. In addition, *ActivityHasSelectResource* is used in *OPL* to decide which unary resource will be selected for processing operation o of locomotive k and using this command to know which run r will be selected by locomotive k to start first. Also, each operation *requires* a unary resource to be processed on it. In this section we will provide a summary for a sugarcane transport system model by CP in the OPL language environment.

Notations:

K number of locomotives.

k,k′ index of locomotives; $k = 1, 2, ... K.$, $k' = 1, 2, ... K$.

E maximum number of segments.

w index of the segments; $w = 1, 2 ... E.$

S_w total number of sections at segment w.

s_w number of sections of segment e; $s_w = 1_w, 2_w, ... S_w.$

$B_{k_\varepsilon o_w s_w}$ number of full bins collected from siding s by locomotive k during operation o and run ε on segment w.

$\alpha_{k_\varepsilon o_w s_w}$ number of empty bins delivered for siding s by locomotive k during operation o and run ε on segment w.

A_{s_w} total allotment of siding s per day.

C_{k_y} capacity of locomotive k of empty bins.

C_{k_c} capacity of locomotive k of full bins.

$\varepsilon, \varepsilon'$ index of runs of each locomotive; $\varepsilon = 1, 2, 3, ..., R$., $\varepsilon' = 1, 2, 3, ..., R$.

$t_{k_\varepsilon o_w s_w}$ start time of locomotive k in run ε on section s on segment w.

$g_{k o_w s_w}$ processing time of operation o of locomotive k on section s on segment w.

C_{max} makespan

Minimise C_{max} (1)

Equation (2) ensures that operation $(o+1)$ of locomotive k cannot be processed before finishing operation o for locomotive k on section s of segment w.

$$t_{k_\varepsilon o_w s_w} + g_{k o_w s_w} \leq t_{k_r (o+1)_w s_w} \quad \forall\, k, \varepsilon, o, s, w. \tag{2}$$

Segments Order

Equation (3) ensures segment w is processed before segment w' or, segment w' before segment w

$$(t_{k_\varepsilon O_w s_w} + g_{kO_w s_w}) \leq t_{k_\varepsilon o_w s'_{w'}} \vee (t_{k_\varepsilon O_{w'} s'_{w'}} + g_{kO_{w'} s'_{w'}}) \leq t_{k_\varepsilon o_w s_w} \quad \forall k, w, w'; w \neq w', \varepsilon, o, s. \tag{3}$$

Selecting Run:

Equation 4ensures that locomotive k selected run ε (unary resource) to be processed.

$$ActivityHasSelect\text{Re}source\ (k,\ Trip,\ \varepsilon) \Leftrightarrow k_\varepsilon \quad \forall\, k \in K,\ \varepsilon \in R. \tag{4}$$

Runs Order

Equation (5) ensures that run ε is assigned for locomotive k before run ε', or run ε' assigned for locomotive k before run ε .

$$t_{k_\varepsilon O_E s_E} + g_{kO_E s_E} \le t_{k_{\varepsilon'} o_w s'_w} \vee t_{k_{\varepsilon'} O_E s'_E} + g_{kO_E s'_E} \le t_{k_\varepsilon o_w s_w}$$
$$\forall k \in K, s, s' \in S, o \in O, w \in E,\ \varepsilon, \varepsilon' \in R;\ \ \varepsilon \ne \varepsilon'. \quad (5)$$

Passing Priorities Constrains

Constraints (6) ensures Locomotives k and k' are processed on section s of segment w. Locomotive k precedes locomotive k' where locomotive k' can't use this segment before locomotive k finishes it, or Locomotives k' and k are processed on section s on segment w and locomotive k' precedes locomotive k where locomotive k can't use this segment before locomotive k' finishes it.

$$t_{k'_\varepsilon o'_w s_w} \ge t_{k_\varepsilon o_w s_w} + \sum_{o=1}^{O_w} \sum_{s=1}^{S_w} g_{ko_w s_w} \vee t_{k_\varepsilon o_w s_w} \ge t_{k'_\varepsilon o'_w s_w} + \sum_{o'=1}^{O_w} \sum_{s=1}^{S_w} g_{k' o'_w s_w}$$
$$\forall k, k' \in K, k \ne k', s \in S_w, w \in E,\ \varepsilon, \varepsilon' \in R, o,\ o' \in O. \quad (6)$$

Makespan Constraint

The following equation ensures each finish each operation for all locomotives during all runs has to be less than Makespan: see equation (7)

$$\max(o \textit{ in operations})\ (t_{k_\varepsilon o_w s_w} + g_{ko_w s_w}) = C_{\max}$$
$$\forall k \in K, s \in S, w \in E,\ o \in O, \varepsilon \in R. \quad (7)$$

Locomotive Constraints

Equations (8) and (9) are locomotive capacity constraints.

In equation (8), the number of empty bins delivered for all sidings during run ε has to be less than or equal to the capacity of locomotive k.

$$\sum_{w=1}^{E} \sum_{o=1}^{O} \sum_{s=1}^{S_w} \alpha_{k_\varepsilon o_w s_w} \le C_{K_y} \ \ \forall k, \varepsilon. \quad (8)$$

In equation (9), the number of full bins collected from all sidings during run ε has to be less than or equal to the capacity of locomotive k.

$$\sum_{w=1}^{E} \sum_{o=1}^{O} \sum_{s=1}^{S_w} B_{k_\varepsilon o_w s_w} \le C_{K_c} \ \ \forall k, \varepsilon. \quad (9)$$

Equation (10 - 11) ensures the delivering has to be in the outbound direction where o and o' are two operations at each siding in the two directions and o' precedes o, so:

If $o' \le o$ then

$$\alpha_{k_\varepsilon o'_w s_w} \le C_{K_y} \ \ \forall k, \varepsilon, o', w, s. \quad (10)$$

$$B_{k_\varepsilon o'_w s_w} = 0 \ \ \forall k, \varepsilon, o', s, w.$$

(11)

Else

If $o' > o$ then

$$B_{k_\varepsilon o'_w s_w} \le C_{K_c} \ \ \forall k, \varepsilon, o', s, w. \quad (12)$$

$$\alpha_{k_\varepsilon o'_w s_w} = 0 \quad (13)$$

End if

Equations (12- 13) ensure the collecting full bins has to be in the inbound direction where o and o' are two operations at each siding in the two directions and o' precedes o.

Bins Constraints

Equations (14) and (15) show the relation between the total allotments for each siding and the number of empty and full bins delivered to or collected from each siding. These constraints will reduce the total number of bins used in the transport process.

In equation (14), the total number of empty bins delivered to siding s has to be equal to the allotment of this siding.

$$\sum_{k=1}^{K} \sum_{\varepsilon=1}^{R} \sum_{w=1}^{E} \sum_{o=1}^{O_w} \alpha_{k_\varepsilon o_w s_w} = A_{s_w} \ \ \forall s. \quad (14)$$

In equation (15), the total number of full bins collected from siding s has to be equal to the allotment of this siding.

$$\sum_{k=1}^{K} \sum_{\varepsilon=1}^{R} \sum_{w=1}^{E} \sum_{o=1}^{O_w} B_{k_\varepsilon o_w s_w} = A_{s_w} \ \ \forall s. \quad (15)$$

Siding capacity:

Equation (16) ensures that the total number of empty and full bins at each siding is less than the capacity of the siding.

$$\alpha_{k_\varepsilon o_w s_w} + B_{k_\varepsilon o_w s_w} \le C_{s_{s_w}} \quad (16)$$

* In CP we cannot use the binary decision variables so, variable subscripts are used as the binary variables. In our model, two variable subscripts are used; s_w and k_ε.

* During implementing the CP code by using OPL software, some commands can be used as well by the following form under constraint programming fundamentals.

- $g_{ko_w s_w}$ variable will be defined as Duration *[k, w, o, s]*.
- The variable $t_{k_\varepsilon o_w s_w}$ can be formulated by constraint programming as task [k, ε, w, o ,s].start

3. Model Solution method:

The previous model is more complicated and need a hug memory and much time to be implemented. The main idea for solving this model is dealing with the sugarcane rail transport system as job shop scheduling problem (Burdett and Kozan (2008, 2010)) and using the constraint programming techniques to solve it (Russell and Urban (2006); (Schaerf, 1999; Henz et al. 2004).The main solving technology of CP is a constraint satisfaction, which deals with problems defined within the finite set of possible values of each variable. This set is often called the domain of the variables. Most of all constraints of industrial applications use a finite domain. The constraint satisfaction problems are solved by using two main techniques: problem reduction and search techniques. Problem reduction is often referred to as consistency maintenance in the literature. This includes node consistency (NC), arc consistency (AC) and path consistency. Many efficient techniques have been developed to achieve AC and NC Constraints (Tsang 1993) and are used for eliminating values from a variable's domain that do not satisfy the constraints.

Many strategies are followed to find the solution through the search trees. The main advantage of these methods is obtaining the optimal solution for many NP-hard problems but on the other side, using some of these techniques individually is time consuming and needs a huge memory to solve large scale problems. Depth-First Search Strategy (DFS) starts at the root node and proceeds by descending to its first descendant. Continuing in this process to reach a leaf. Then we backtrack to the parent of the leaf and descend to its next descendant, until it exists (Tsin, 2002, Her and Ramakrishna 2007). Best-first search (BFS) is a well-known strategy for optimization problems that consists of choosing the node with the best value of the objective function. For each job shop scheduling problem, we set the initial upper-bound on the Makespan to be the sum of the durations of all activities. If a solution is found with a lower Makespan, the new bound is added at the leaf node. Sierra and Varela (2008) used a Best First Search to reduce the search domain in solving the job shop scheduling problem with total flow time. Interleaved depth first search (IDFS) works through DFS where IDFS prevents DFS falling into mistakes. While DFS searches sequentially from left to right, IDFS searches in parallel many subtrees at some tree level which is called active. It searches in each subtree to reach each leaf. If this leaf is the goal, the search process terminates. Otherwise, the state of the current subtree is recorded and added to a list of active subtrees. Slice Based Search (SBS) and Depth-bound Discrepancy Search (DDS) are effectives for job shop problems(Walsh, 1997; Beck &Perron, 2000) and will be used in this study.

Standard Strategy (SSS)and Dichotomic Search Strategy(DSS) will be integrated with the pervious search techniques inside OPL to improve the profermance of the previous search techniques. In SSS the objective function is evaluated at the first feasible solution point, and an upper bound is obtained and then the constraints are added to obtain the next solution. DSS depends on evaluating the objective function at the first feasiable solution and at the mid value.

Algorithm1: Delivering and Collecting techniques:

Algorithm 1 includes the procedures which are followed to deliver and collect bins during each run for each locomotive. From the model, we assume that there are two operations o, o′ that will be implemented on each section during the locomotive run r. The first operation will be executed during the outbound direction and the second operation will be in the inbound direction. In addition, algorithm 1 considers no delivering or collecting if the section is a passing section and doesn't work as a siding for delivering or collecting bins.

This algorithm assigns first operation for delivering bins only during outbound direction and no collecting occurs for reducing the operating cost of locomotive and sometimes for safety conditions (no fulls and empties at the same time in the locomotive).In this algorithm, the delivering will be with the smaller operation. The second operation on the same section and the same locomotive will be assigned for collecting full bins during inbound direction of the locomotive. This algorithm supposes there is no delivering in the inbound direction. From the algorithm, the collecting will be associated to the greater operation.

Delivering and collecting procedure:

1. **Select** Loco k; k ϵ K.
2. **Select** section s; s Є S
3. **If** s is a siding **then**
4. **If o′ ≤ o then** // inbound direction

   $$\alpha_{k_\varepsilon o'_w s_w} \le C_{K_y}$$

   $$B_{k_\varepsilon o'_w s_w} = 0$$

   $$\alpha_{k_\varepsilon} = \alpha_{k_\varepsilon} - \alpha_{k_\varepsilon o'_w s_w}$$

   go to step 2
5. **else** // outbound direction

   $$B_{k_\varepsilon o'_w s_w} \le C_{K_c}$$

   $$\alpha_{k_\varepsilon o'_w s_w} = 0$$

   $$B_{k_\varepsilon} = B_{k_\varepsilon} - B_{k_\varepsilon o'_w s_w}$$

   go to step 2
6. **Else**
7. end if
8. **Else**

$$\alpha_{k_\varepsilon o'_w s_w} = 0$$

$$B_{k_\varepsilon o'_w s_w} = 0$$

go to step 2

9. End if;

10. End.

4. Computational Results:

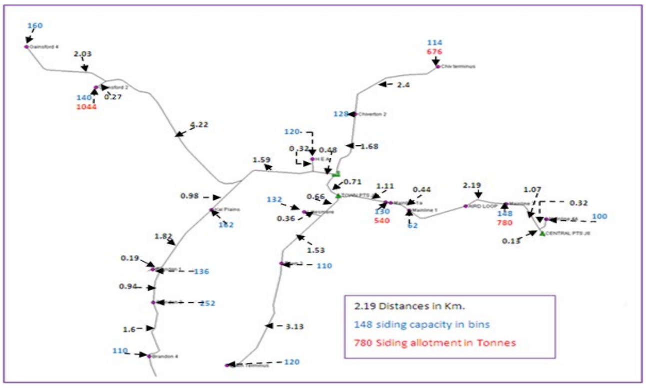

Fig. 1 a realistic test case study: small sector of rail network of Kalamia mill

To demonstrate and validate the model in this research, the real test case study will be used. A sector of the transport system at Kalamia mill, near Ayr south of Townsville, will be used to obtain the optimal completion time of all runs of locomotives per day. In this case just we used a small sector of the whole transport network. Fig. 1 shows some information about the distances between sidings, sidings' capacity and allotments of sidings. Our test case study involves 26 sections, 15 sidings and 3 locomotives. OPL software is used to obtain a solution for our case study. The CP formulation model is solved by the OPL solver by using constraint programming search techniques and the ILOG scheduler. SSS and DSS are applied for the model as well to obtain solutions for the CP model.

The Comparison between SSS and DSS in table 1 clarifies that all techniques have given results in a more reasonable and shorter time under DSS than when using SSS. DFS has achieved a good CPU time with the two strategies, while the BFS gave shorter times with DSS, whoever when combined with SSS, it was unable given any result. As a result, we can consider the DFS technique is also suitable for solving the CP formulation.

Table1: Comparison of SSS and DSS strategies

Search technique	SSS		DSS	
	Solution	CPU Time	Solution	CPU Time
DFS	2664	100.68	2664	17.81
SBS	2664	101.34	2664	18.06
DDS	2664	102.61	2664	19.31
BFS	na	na	2664	11.41
IDFS	2664	106.64	2664	18.16

Locomotives and Runs Scheduling for CP Model:

Table 2 shows the exact start times and finish times of the locomotives' run and assigns the runs for locomotives. From Table 4, locomotives 1, 3, and 4 start at time 0 and are scheduled first, second and third respectively, but all have the same priority to start. Locomotive 5 is the fourth, and locomotive 2 is the fifth. For the CP model results, three locomotives can start at the same time for our case at time zero.

Table 2: Start and finish times of locomotives' run for CP model.

index of locos	Index of runs	Siding number	Visit time	empties	fulls	Run time	Start time	Finish time
1	1	6	791	120	120	2414	0	1582
2	5	6	1873	54	54	1582	250	2664
3	2	14	491	112	112	982	0	982
4	3	20	220	90	90	1890	0	1976
		22	537	10	20			
5	4	22	1439	120	110	1074	86	2240
Total	5	-	-	506	506	7942	0	2664

All locomotives have to collect the sidings' allotments in that day and deliver the same number of empty bins for each siding. Table 2 shows the actual visit times for all locomotives at each siding, the number of sidings which will be visited, the amount of empty and full bins which will be delivered or collected from each siding and locomotives' run times. Locomotive's run time means the time which each locomotive spends between starting at the run at the mill and returning to the mill after visiting the sidings. Figure 2 show the locos scheduling where segments 8 and 10 include two directions outbound direction and inbound directions respectively which will be implemented on the same segment 8.

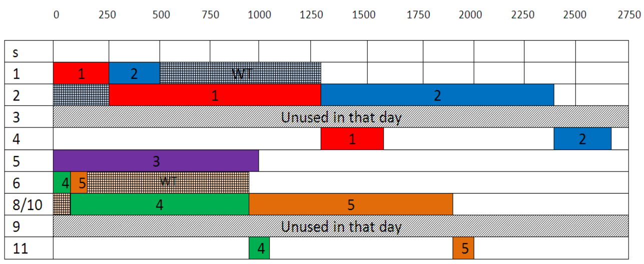

Figure2: Gantt chart for scheduling 5 locomotives on 11 segments with the Makespan of 2664

Solutions Analysis by Search Tree:

After applying the CP search techniques for the model, in this section we try to explain how OPL has used the search tree techniques to obtain case study test solutions and how the search tree is built by branching techniques. For our test case study, by using the CP model, for instance, we have obtained four solutions before obtaining the optimal solution by using DFS with SSS. Table 3shows all solutions and CPU time to obtain each solution.

Table 3: Solution analysis under DFS and SSS

Solutions	Nb. of variables	Nb. of constraints	Nb. of choice points	Nb. of failure points	Solving time (s)	Makespan
First sol.	30253	380150	12683	0	9.48	3356
Second sol.	30253	380150	13237	554	11.02	3106
Third sol.	30253	380150	25915	13234	98.7	2914
Fourth sol.	30253	380150	26469	13788	99.9	2664
Optimal sol.	30253	380150	26469	26466	100.68	2664

Conclusion:

The proposed scheduling model is very complicated and needs to be solved in a reasonable time, because of the dynamic nature of the system and large number of variables. The CP formulation is used to solve the sugarcane rail transport problem. CP works well with all five search techniques under the two search strategies, except BFS with SSS. In addition, the DSS positively affected the CPU time for all search techniques after and before applying the algorithms.

REFERENCES

Beck, J.C., Perron, L.(2000) Discrepancy-Bounded Depth First Search. In Proceedings Of the Second International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR 2000). Cited on page 21, 22.

Burdett, R.L., Kozan, E.(2008) A sequencing Approach for Creating New Train Timetables. *OR Spectrum*, DOI 10.1007/s00291-008-0143-6

Burdett, R.L., Kozan, E.(2010) A disjunctive Graph Model and Framework for Constructing New Train Schedules. *European Journal of Operational Research* 200:85–98.

Henz M, Müller T, Thiel S.(2004) Global Constraints for Round-Robin Tournament Scheduling. *European Journal of OperationalResearch,* 153(1):92–101.

Her, J.H., Ramakrishna,R.S.(2007) An external-memory depth-first search algorithm for general grid graphs. *Theoretical Computer Science,* 374: 170-180.

Masoud M, Kozan E, Kent G (2010) Scheduling techniques to optimise sugarcane rail systems. ASOR Bulletin 29:25-34

Russell, R. A., Urban, T. L.(2006) A constraint Programming Approach to The multiple-Venue, Sport-Scheduling Problem. Computers & Operations Research 33: 1895-1906.

Schaerf, A.(1999) Scheduling Sport Tournaments Using Constraint Logic Programming.

Constraints, 4(1):43–65.

Sierra, M. R., Varela, R.(2008) Pruning by Dominance in Best-First Search for The job Shop Scheduling Problem with Total Flow Time. *J Intell Manuf.* DOI 10.1007/s10845-008-0167-4.

Tsang, E.(1993) Tsang, Foundations of Constraint Satisfaction, Academic Press, London and San Diego (1993).

Tsin, Y.H.(2002) Some remarks on Distributed Depth-First *Search. Information Processing Letters* 82: 173–178.

Walsh, T. (1997)Depth-Bounded Discrepency Search. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pp. 1388–1395. Morgan Kaufmann

AUTHORS BIOGRAPHIES

Mahmoud Masoud is a PhD student at QUT, Australia. He had a master in Operations Research at Cairo university-Egypt, 2004. He works as an assistant lecturer at Cairo University.