An Aggregated Rank Removal Heuristic Based Adaptive Large Neighborhood Search for Work-over Rig Scheduling Problem

Naveen Shaji(✉), Cheruvu Syama Sundar, Bhushan Jagyasi, and Sushmita Dutta

Accenture Advanced Technology Centers in India, Mumbai, India
{naveen.shaji,syama.sundar.cheruvu,bhushan.jagyasi, sushmita.dutta}@accenture.com

Abstract. Work-over Rig Scheduling Problem (WRSP) is a well known challenge in oil & gas industry. Given the limited number of work-over rigs to cater to the maintenance needs of a large number of wells, the challenge lies in planning an optimum schedule that minimizes the overall production loss. In this work, we propose a new Aggregated Rank Removal Heuristic (*ARRH*) applied to Adaptive Large Neighborhood Search to solve WRSP. The proposed approach results in more efficient searches as compared to existing heuristics - Genetic Algorithm, Variable Neighborhood Search and Adaptive Large Neighborhood Search.

1 Introduction

Oil wells require various maintenance services from time to time. Work-over rigs are specialized mobile units that are designed to carry out these maintenance activities. However rigs are expensive units and hence limited in number. The wells that require maintenance, wait in queue for a rig to visit them and carry-out the needed work-over, all the while remaining idle and resulting in production loss. Hence, there is a need to generate an optimum maintenance schedule that minimizes the production loss. The problem of optimizing schedule for work-over rigs is combinatorial optimization problem with constraints and belongs to the NP-Hard difficulty. As the number of wells to be serviced increases, the size of the problem increases exponentially which can cause the execution times of these searches to go beyond an acceptable time.

The Work-over Rig Scheduling Problem (WRSP) can be seen as a type of the Vehicle Routing Problem (VRP) [3,10] but with the objective to minimize the waiting time for the wells multiplied by the production loss. There have been several solutions proposed for solving the WRSP using various heuristics which has been summarized in [7]. Aloise et al. [1] employed Variable Neighborhood Search (VNS) heuristic to solve WRSP. Ribeiro et al. [8] compares between three meta heuristics, Iterated local search (ILS), Clustering search (CS) and Adaptive

https://doi.org/10.1007/978-3-030-34869-4_42

large neighborhood search (ALNS). Clustering search proceeds by assigning the solutions obtained by the search to various clusters based on minimum hamming distance. The authors claim that the best results were obtained with ALNS.

Several recent works [4–6,9] deals with Capacitated Cumulative Vehicle Routing Problem (CCVRP) which shares a close analogy with WRSP. Ngueveua et al. [6] is the first to formulate CCVRP, draw insights into why CCVRP is tougher than CVRP and uses Memetic Algorithm (Genetic Algorithm (GA) supported with local search) to solve it. Lysgaard et al. [5] uses the Branch-and-Cut-and-Price algorithm to solve CCVRP. Sze et al. [9] and Liu et al. [4] proposes hybrids of the Large Neighborhood Search to solve CCVRP and highlights the robustness and efficiency of the approach.

Since ALNS is one of the state of the art algorithms for solving WRSP, our work focuses on improving ALNS for achieving better solutions in a quicker time. The key contribution of this paper is a new Aggregated Rank Removal Heuristic (*ARRH*) based ALNS to solve the WRSP. We compare the proposed *ARRH* based ALNS with existing ALNS to show the performance benefits in terms of production loss.

2 Problem Definition

In this section, the work-over rig scheduling problem (WRSP) and its constraints is formulated mathematically. Different wells require different levels of maintenance which can only be serviced by a rig having a equal or higher level work-over capability. To model WRSP, it is assumed that we know before hand the wells that require work-over, their production loss and various travel times between the wells. A good model of the problem and its constraints has been provided by Ribeiro et al. [8], which has been used in our work.

Let W be the set of wells to be scheduled and K be the set of available rigs. Every schedule s is an association of each rig $k \in K$ to a directed graph, $G^k = (V^k, A^k)$, where V^k is the set of nodes and A^k is the set of arcs. Let W^k be the set of wells assigned to rig k then $V^k = W^k \cup \{o_k, z_k\}$, where o_k is the node representing the starting position of rig k, $k \in K$, z_k is the node representing the ending position of rig k, $k \in K$. Every arc $(i,j) \in A^k$ is characterized by a time duration $c_{ij} = e_{ij} + d_j$, where e_{ij} is the travel time between nodes i and j , $i,j \in W \cup_{k \in K} \{o_k\}$ and d_j is the duration of the maintenance (work-over time) required by node j, $j \in W$. In case of missing paths between any two nodes, travel time will be calculated as the minimum spanning time with one or more intermediate nodes.

We assume dummy arcs (i, z_k), $i \in W^k$ with zero duration. Also $i \in W^k \cup \{o_k\}$ & $j \in W^k$ $\forall (i,j) \in A^k$.

To denote the presence or absence of an edge (i,j) in schedule of rig k we use a binary variable y^k_{ij} as given by Eq. 1.

$$y^k_{ij} = \begin{cases} 1 & if\ (i,j) \in A^k, \\ 0 & otherwise. \end{cases} \tag{1}$$

We now formulate $x_j(s)$, the time waited by the node j in the queue for the maintenance to begin in schedule s, using Eq. 2.

$$for \;\; s \; s.t \;\; j \in A^k, \;\; x_j(s) = \begin{cases} e_{o_k j} & if \; y^k_{o_k j} = 1 \\ x_i(s) + d_i + e_{ij} & if \; y^k_{o_k j} = 0 \;\; and \;\; y^k_{ij} = 1 \end{cases} \tag{2}$$

Objective Function: The objective function Ψ to be minimized is given by Eq. 3.

$$\Psi(s \mid s \vdash G^k, \; k \in K) = \sum_{k \in K} \sum_{j \in W^k} (x_j + d_j) p_j \tag{3}$$

where, p_j is the oil production rate of node j, $j \in W$.

Constraints

1. All wells demanding maintenance must be present in the combined schedule of all rigs. That is $W = \sum_{k \in K} \cup W^k$.
2. Each well is visited exactly once in the combined schedule of all rigs.

$$\sum_{k \in K} \sum_{j:(i,j) \in A^k} y^k_{ij} = 1, \;\; i \in W \tag{4}$$

3. The classic network flow constraints are modeled as.

$$\sum_{j:(o_k,j) \in A^k} y^k_{o_k j} = \sum_{i:(i,z_k) \in A^k} y^k_{i z_k} = 1, k \in K \tag{5}$$

$$\sum_{j:(i,j) \in A^k} y^k_{ij} - \sum_{j:(j,i) \in A^k} y^k_{ji} = 0, \;\; k \in K, \; i \in W^k \tag{6}$$

4. Level Constraint: Level constraint is modeled to make sure that wells are visited by those rigs that are equipped to serve the level of maintenance demanded.

$$q_k \geq l_j \;\; \forall \, j : (j, i) \in A^k, \; k \in K \tag{7}$$

where, l_j is the level of maintenance request at node j, $j \in W$ and q_k is the maximum level of maintenance request that can be attended by rig k.

3 Algorithms

3.1 Genetic Algorithm Supported Variable Neighborhood Search (VNS+GA)

We provide below a brief description of the implemented GA and VNS Algorithms along with the hybrid GA supported VNS.

Genetic Algorithm. The initial population was created by means of iteration through each rig multiple times appending a random well (not yet assigned), to its schedule till all requests for maintenance were met by the schedule.

Crossover- In this work, to cross between two schedules the following approach was used.
Let PA, PB denote the individuals to be crossed.

1. All wells that belong outside crossover point is removed from PA to form PA'.
2. All wells that are in PA' are eliminated from PB forms PB'.
3. Concatenation of PB' to PA' retaining the structure gives child C.

Mutation- In this work, mutation implements two variation in the child. A random interchange of wells within the schedule of a rig and a random well being shifted from one rig's schedule to another. The number of children to be mutated are given by a user defined parameter τ. Mating Pool- In this work the method of Roulette wheel selection [2] is used. The number of children is given by a user defined parameter n_{child}.

Variable Neighborhood Search (VNS). Our attempt to solve the WRSP using VNS was inspired and extends the work of Aloise et al. [1] which employed VNS heuristic to solve WRSP. The VNS framework adopts a recursive search involving an exhaustive search of local neighborhoods and a random search of global neighborhoods (a neighborhood is a set of possible schedules that are similar to a given schedule in some way).

For a given schedule its neighborhoods are generated using the following operations

1. Swap Wells from Same Work-over rig ($SWSW$): In the given schedule, swap two wells assigned to same rig.
2. Swap Wells from Different Work-over rigs (SWDW): In the given schedule, two wells assigned to two different rigs respectively are interchanged.
3. Add Drop (AD): In the given schedule, a well assignment to one rig is dropped and it the well is assigned to another rig.

The local neighborhoods considered are SWSW, SWDW and AD. The global neighborhoods are SWSW done twice and thrice, SWDW done twice and thrice and the AD done twice and thrice.

Genetic Algorithm Supported Variable Neighborhood Search. In Genetic Algorithm Supported Variable Neighborhood Search a GA is employed by the VNS heuristic to carry out its local search routine. From the current schedule several solutions in the neighborhood defined by the VNS are created and these solutions become the initial population of GA. The idea here is to remove the in-efficient exhaustive search and replace it by the much faster GA.

3.2 Proposed *ARRH* Based ALNS

In this section, we first briefly describe the existing ALNS implementation, and then present our proposed *ARRH* based ALNS.

Adaptive Large Neighbourhood Search (ALNS). Our work is inspired from and extends the work of Ribeiro et al. [8] which highlights the superiority of ALNS. Below we provide a brief description of ALNS, a detailed view is available in Ribeiro et al. [8]. ALNS proceeds by destroying a part of the existing solution and recreating it. In each iteration it removes a certain number wells from the schedule and then inserts it back into better positions. For this purpose, it employs various insertion and removal heuristics. The probability of selection of a removal or insertion heuristics depends on its past performance in the search.

Objective Function- The objective function in ALNS is modified to add an additional term λ for the broken level constraints. Here λ is a user defined parameter.

The Proposed Aggregated Rank Removal Heuristic Based ALNS. Aggregated Rank Removal Heuristic (*ARRH*) which is proposed to replace the Worst Removal Heuristic (*WRH*) in ALNS. The general idea here is that although ALNS is selecting wells based on travel times and production losses using separate methods, such methods lack the power to make unbiased identification of opportunities that arise due to a combination of these factors. The *ARRH* is designed to solve two shortcomings of the WRH as given below.

1. Bias Factor - For optimization, the general idea behind removal heuristics is to remove those wells that has an opportunity to be placed in a better position. However, evaluating the wells solely based on loss has a tendency to give more preference to those wells that have higher production rate. Similarly large wait times can also occur from wells with large work-over time or a well isolated from others, inducing a large travel time. Hence larger production loss doesn't always imply the presence of an opportunity for the well to be placed in a better position.
2. Masking Factor - The WRH takes into consideration of the contribution to production loss only. A well's contribution to over all production loss is driven not only by its own maintenance requirement but also because of the previous wells that make the well under consideration to wait for its turn. However these other wells' contributions are masked in WRH.

Proposed ARRH- The proposed approach, Aggregated Rank Removal Heuristic (*ARRH*) works on ranked values of (a) production loss, (b) travel times and (c) work-over time. Ranked values corresponding to each metric is created by ordering the wells in the increasing order of the metric. Then a discrete set of ordered values ranging from 0 to 1 of equal spacing is created and assigned to the wells as their ranked value of the metric. Ranking the metric in such fashion will help identify opportunities for a better solution without bias.

Henceforth, in this paper, the ranked values of travel times, work-over time, production rate and ratio of production rate to work-over time will be represented by $\hat{e}_{i,j}$, $\hat{d}_j$, $\hat{p}_j$ and $\hat{r}_j$ respectively. *Scoring* - Scoring the well based on multiple criteria can help pinpoint the cause of the increased production loss. *ARRH* removes wells with the highest scores. In this work, three criteria have been used to determine the goodness of a solution.

– Suffocation - If any rig is overloaded with many wells with high production losses and large work-over times no matter the ordering, the wells assigned to the rig will face large production losses, indicating that some of the wells assigned to the rig needs to be removed and placed in the schedule of another rig. The suffocation coefficient for a rig is calculated by Eq. 8.

$$\Gamma(k) = \sum_{j \in W^k} \hat{p}_j + \hat{d}_j \tag{8}$$

– Ordering - Ordering criterion measures how the wells are ordered in the schedule of a rig. It follows the same criteria as described in Bassi et al. [11]. The ordering coefficient for a rig is calculated by Eq. 9.

$$\zeta(k) = \frac{4}{n(n+1)} \sum_{j \in W^k} i\ \hat{r}_j \tag{9}$$

Where i = 0, . . ., n is the numbering of wells in schedule of rig r.

– Travel-time - The suffocation and ordering criteria deals with the distribution and arrangement of rigs based on $\hat{p_j}$ and $\hat{d_j}$. An optimization based only on these criteria alone may lead to coupling between wells that are far apart and induce large travel times rendering the entire procedure in-feasible. The travel time criterion mitigates this by evaluating the travel times at the edges of each well. The travel time coefficient is given by the Eq. 10.

$$\xi(i) = \begin{cases} 2e_{i,\hat{i+1}} & if\ y_{o_k i} = 1 \\ 2e_{i,\hat{i-1}} & if\ y_{iz_k} = 1 \\ e_{i,\hat{i+1}} + e_{i,\hat{i-1}} & Otherwise \end{cases} \tag{10}$$

Where $i \in W^k$ and $(i, i+1)$, $(i-1, i) \in A^k$

In order to give an equal importance to all criteria, the coefficients by the design of the equations are set to vary between 0 and 2.

The criteria mentioned above reflects on some trends we want to see in the schedule however, they are not mutually exclusive. A schedule that is suitable to one criteria need not suit another. Therefore there is a need to strike a balance between them using stochastic aggregation.

In *ARRH* wells are removed based on a score given by Eq. 11.

$$s(i) = x\Gamma(k|i \in k) + y\xi(i) + z\zeta(k|i \in k) \tag{11}$$

where i is the well,
r is the rig to which i belongs in the current solution,
x, y, z are random numbers between 0.8 and 1.

A time complexity comparison between $ARRH$ and WRH shows that they both have similar average time complexities $O(|W|^2)$, where $|W|$ is the number of wells that needs to be serviced. Note that that time complexity for both the heuristics is not dependent on the number of rigs.

4 Results

4.1 Experimental Setup

The data used to create different instances of the problem were created using random simulations. All algorithms were implemented using python 3, on a Windows 10 Enterprise 64 Bit system with Intel(R)Core(TM) i7-8650U CPU @ 1.90 GHz and available physical memory of 9.82 GB. For readability and ease of comparison all losses reported are scaled down by a factor of 5000.

The parameters were created as:

- Production loss (p_j)- Production loss of each well is sampled from a uniform distribution ranging from 20 to 60.
- Work-over times (d_j)- Work-over times were sampled from a uniform distribution ranging from 20 to 80.
- Travel times ($e_{i,j}$)- x and y co-ordinates were sampled from a uniform distribution ranging from 0 to 100. Travel times were calculated as the euclidean distance between two points.

4.2 A Brief Comparison of GA, VNS+GA and ALNS

All three algorithms were tried to solve an instance where VNS failed to deliver results in a feasible time. That is, for 50 wells and 10 Rigs, VNS gave a loss of 78.8956 for an epoch taking 9138 s. Hence we had an estimated 130 h for 50 epochs. For 5 runs, ALNS, VNS+GA and GA gave 70.04, 78.23, 110.51 average losses respectively, for a run time of 2000 seconds. The parameters used for the algorithm are given in Table. 1. The Fig. 1(a) compares the production loss with respect to the run time of these three algorithms.

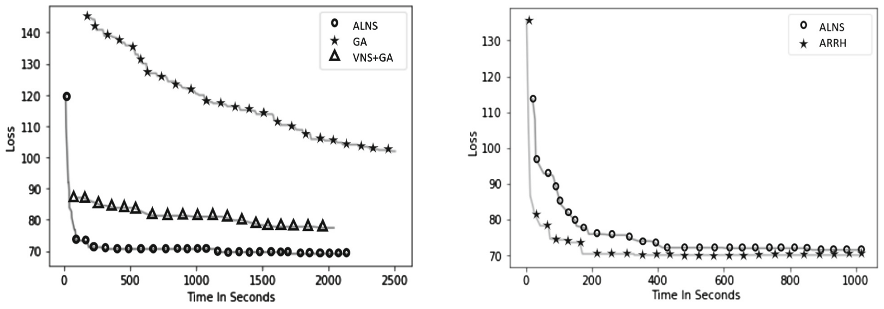
4.3 Comparison of ALNS Vs *ARRH* Based ALNS

ALNS and $ARRH$ based ALNS were compared for 18 instances and the results are tabulated in Table. 2. All instances were run 5 times repeatedly to decrease the uncertainty in the result. The entries made into the table are:

- Instance: Specifies the instance, the n/m notation stands for n wells and m rigs.

Table 1. User defined parameters

Heuristic	Symbol	Meaning	Value
GA	n_{pop}	Population Size	700
	n_{child}	Number of Children	600
	τ	Rate of Mutation	13
VNS+GA	n_{pop}	Population Size	20
	n_{child}	Number of Children	16
	τ	Rate of Mutation	2
ALNS and M_ALNS	ϵ	Controls Number of wells removed	0.4
	ϕ	Number of iterations in a segment	50
	λ	Penalizes broken level constraints	15
	ν	Avoids determinism in Shaw removal heuristics	2
	$\varPhi$	Avoids determinism in Route removal heuristic	3
	ρ	Avoids determinism in Worst removal heuristic	2
	T_{start}	Starting temperature in Simulated Annealing	15
	c	Temperature reduction factor	0.995



(a) Comparison between ALNS, GA supported VNS and GA.

(b) Comparison between ALNS and *ARRH* based ALNS

Fig. 1. Loss vs time comparison

– Time: To be fair both algorithms were run for the same time for an instance and is mentioned in the table in seconds.
– Average Loss: Gives the average loss over 5 times.
– Best: Gives the best-found solution from the 5 times.
– Deviation: Here deviation is defined as $\frac{AverageLoss-Best}{Best} \times 100$.

Figure 1 provides a comparison between ALNS and proposed *ARRH* based ALNS. As seen from the graph, proposed *ARRH* based ALNS results in improvement in production losses as compared to ALNS for the same run time. Further if there is a benchmark of production losses *ARRH* based ALNS is able to achieve it faster.

Table 2. ALNS Vs *ARRH* based ALNS

Instance	Time (s)	ALNS			*ARRH* based ALNS		
		Average	Deviation	Best	Average	Deviation	Best
50/5(1)	250	123.44	1.59	121.47	119.53	1.69	117.50
50/5(2)	250	126.12	3.53	121.66	123.46	2.10	120.86
50/5(3)	250	139.07	1.86	136.47	132.12	5.22	125.22
50/10(1)	250	74.91	3.63	72.19	73.48	1.97	72.03
50/10(2)	250	78.23	3.50	75.49	74.65	1.55	73.49
50/10(3)	250	77.45	4.78	73.75	73.95	2.71	71.94
100/5(1)	3600	494.75	1.31	488.24	472.08	2.39	460.78
100/5(2)	3600	450.97	2.22	440.94	442.14	1.19	436.86
100/5(3)	3600	395.22	3.18	382.63	390.52	3.51	376.81
100/10(1)	3600	246.63	1.80	242.17	236.45	1.53	232.83
100/10(2)	3600	260.58	0.73	258.65	257.27	0.99	254.70
100/10(3)	3600	236.34	1.17	233.57	231.89	0.68	230.29

ALNS and *ARRH* based ALNS were also compared using Manhattan distance to calculate distance between the wells. The results were collected for 5 instances of 20 and 30 wells each and *ARRH* based ALNS was found to give an average improvement of 2.14 % over ALNS. This insight is similar to results obtained from euclidean distance, hence the approach is robust to various distance metrics.

5 Conclusion

The Workover Rig Scheduling Problem (WRSP) has been addressed to improve the production losses of the well. We presented the implementation of the Genetic Algorithm based Variable Neighborhood Search algorithm for the same problem. This combination had a significant improvement than both GA and VNS implementations. However, GA based VNS was not found to be superior than existing Adaptive Large Neighborhood based (ALNS) approach. In this paper, a new heuristic - Aggregated Rank Removal Heuristic (ARRH) had been proposed to be implemented with the existing Adaptive Large Neighborhood Search (ALNS) algorithm for WRSP. The proposed ARRH is based on the utility of rank-based methods to make unbiased judgments about the schedules. The results indicates that this proposed approach results in the lower production losses than ALNS approach for equal run times. In this work, a removal heuristic was designed and used on the basis of ranking. In future we aim to introduce a rank based insertion heuristic as well.

References

1. Aloise, D.J., Aloise, D., Rocha, C.T., Ribeiro, C.C., Filho, J.C.R., Moura, L.S.: Scheduling workover rigs for onshore oil production. Discrete Appl. Math. **154**(5), 695–702 (2006). https://doi.org/10.1016/j.dam.2004.09.021. http://www.sciencedirect.com/science/article/pii/S0166218X05003008. IV ALIO/EURO Workshop on Applied Combinatorial Optimization
2. Goldberg, D.E.: Genetic Algorithm in Search Optimization and Machine Learning. Addison-Wesley, Reading (1989)
3. Li, F., Golden, B., Wasil, E.: The open vehicle routing problem: algorithms, large-scale test problems, and computational results. Comput. Oper. Res. **34**(10), 2918–2930 (2007). https://doi.org/10.1016/j.cor.2005.11.018. http://www.sciencedirect.com/science/article/pii/S0305054805003515
4. Liu, R., Jiang, Z.: A hybrid large-neighborhood search algorithm for the cumulative capacitated vehicle routing problem with time-window constraints. Appl. Soft Comput. **80**, 18–30 (2019). https://doi.org/10.1016/j.asoc.2019.03.008. http://www.sciencedirect.com/science/article/pii/S1568494619301267
5. Lysgaard, J., Wøhlk, S.: A branch-and-cut-and-price algorithm for the cumulative capacitated vehicle routing problem. Eur. J. Oper. Res. **236**, 800–810 (2014). https://doi.org/10.1016/j.ejor.2013.08.032
6. Ngueveu, S.U., Prins, C., Calvo, R.W.: An effective memetic algorithm for the cumulative capacitated vehicle routing problem. Comput. Ope. Res. **37**(11), 1877–1885 (2010). https://doi.org/10.1016/j.cor.2009.06.014. http://www.sciencedirect.com/science/article/pii/S0305054809001725. metaheuristics for Logistics and Vehicle Routing
7. Ribeiro, G., Mauri, G., Lorena, L.: A simple and robust simulated annealing algorithm for scheduling workover rigs on onshore oil fields. Comput. Ind. Eng. **60**, 519–526 (2011). https://doi.org/10.1016/j.cie.2010.12.006
8. Ribeiro, G.M., Laporte, G., Mauri, G.R.: A comparison of three metaheuristics for the workover rig routing problem. Eur. J. Oper. Res. **220**(1), 28–36 (2012). https://doi.org/10.1016/j.ejor.2012.01.031. http://www.sciencedirect.com/science/article/pii/S0377221712000665
9. Sze, J.F., Salhi, S., Wassan, N.: The cumulative capacitated vehicle routing problem with min-sum and min-max objectives: an effective hybridisation of adaptive variable neighbourhood search and large neighbourhood search. Transp. Res. Part B Methodol. **101**, 162–184 (2017). https://doi.org/10.1016/j.trb.2017.04.003. http://www.sciencedirect.com/science/article/pii/S0191261516308396
10. Vidal, T., Crainic, T.G., Gendreau, M., Prins, C.: Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. Eur. J. Oper. Res. **231**(1), 1–21 (2013). https://doi.org/10.1016/j.ejor.2013.02.053. http://www.sciencedirect.com/science/article/pii/S0377221713002026
11. Bassi, H.V., Ferreira Filho, V.J.M., Bahiense, L.: Planning and scheduling a fleet of rigs using simulation-optimization. Comput. Ind. Eng. **63**, 1074–1088 (2012). https://doi.org/10.1016/j.cie.2012.08.001