High-Accuracy Scalable Solutions to the Dynamic Facility Layout Problem

Apan Qasem
Texas State University
apan@txstate.edu

Clara Novoa
Texas State University
cn17@txstate.edu

Chandra Sekhar Kolla
Texas State University
c_k35@txstate.edu

Samantha Coyle
Texas State University
samc@txstate.edu

KEYWORDS

facility layout, parallel computing, performance tuning

ACM Reference Format:
Apan Qasem, Clara Novoa, Chandra Sekhar Kolla, and Samantha Coyle. 2018. High-Accuracy Scalable Solutions to the Dynamic Facility Layout Problem. In *Proceedings of 2018 ACM/IEEE Supercomputing Conference (SC'18).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Between 20% and 50% of the operating expenses in manufacturing in the United States are attributed to material handling and changes to plant layout. Effective facilities planning can reduce these costs by as much as 30%. In the Dynamic Facility Layout Problem (DFLP), the flow of materials between facilities vary over T time periods in the planning horizon due to fluctuations in the final product demands, changes in machinery, etc. The DFLP finds a layout or assignment of n facilities to n plant locations in each period that minimizes the total material handling and relocation costs. The size of the problem search space, $n!^T$, becomes intractable by exact methods even if n and T are as small as 10 and 5, respectively.

2 PROBLEM FORMULATION

In the DFLP, the flow of material between facilities required to produce products or services is known but it changes over the periods. Changes in these flows may increase future material handling costs (MHC) for a given plant layout triggering a need for switching facilities and incurring in relocation costs (RC). The DFLP can be formulated as a non-linear optimization problem with the objective of finding an optimal layout for each period so that the total MHC and RC incurred over the time horizon is minimized as shown in eq. (1).The binary decision variables are notated as x_{tki} and y_{tkij}; x_{tki} is one if facility k is assigned to location i in period t, zero otherwise, and $y_{tkij} = x_{(t-1)il} * x_{til}$ is one if k is shifted from i in $t-1$ to location j in t, zero otherwise. The model parameters are the length of the horizon, T, usually given in years, the number of facilities, n, which is also the number of locations, the flow of material between facilities k and l at time t, f_{tkl}, the distances between plant locations i and j, d_{ij}, and the cost of relocating k from i to j in period t, a_{tkij}. Two sets of constraints, omitted for brevity, enforce the assignment of each facility to a single location and the assignment of each location to one facility in each period.

$$\begin{aligned} min \quad z \quad = & \sum_{t=1}^{T}\sum_{k=1}^{n}\sum_{l=1}^{n}\sum_{i=1}^{n}\sum_{j=1}^{n} f_{tkl} d_{ij} x_{tki} x_{tlj} \qquad (1) \\ + & \sum_{t=2}^{T}\sum_{k=1}^{n}\sum_{i=1}^{n}\sum_{j=1}^{n} a_{tkij} y_{tkij} \end{aligned}$$

3 RELATED WORK

Exact approaches to solve small sized DFLP's include cutting plane [5], dynamic programming (DP) [8], and linear network model [2]. Heuristic and meta-heuristic algorithms such as simulated annealing, tabu search, hybrid ant systems [6], and iterated great deluge [7] have been used for problems with more than 10 facilities. In [4] a robust approach (RA) is suggested as an alternative to DP. In the RA, a robust layout which is not necessarily optimal for a particular period is found and implemented for the whole time horizon. In contrast to DP, the RA avoids relocation costs but no optimal layout is implemented for each period.

4 OUR SOLUTION APPROACH

An arrangement of n facilities to n locations in a particular period t (i.e. a *layout*) is represented using the permutation $\{1, 2, ...n\}$. Its MHC is computed as the sum of the product of all bi-directional flows between the facilities and their distances given by their assigned locations. We model the search space over T time periods as a directed graph in which there are V nodes and each node represents a layout. An edge $E_{(t-1),t}$ between nodes (v_{t-1}, v_t) has a weight equal to the sum of the MHC for v_t and the RC from v_{t-1} to v_t With this representation, solving the DFLP is reduced to finding the all-pairs shortest path (SP) in the periodic layout graph. The problem is converted to a single-source SP by the addition of dummy nodes v_o and v_f at the beginning and end of the planning horizon These dummy nodes have zero MHC associated and zero RC on their outcoming or incoming edges. We employ an augmented version of the Fredman-Tarjan algorithm (FTA) [3] to find the SP. We make the key observation that in the DFLP, selection of a node to label as *marked* does not require traversing the temporary

---

SC'18, November 2018, Dallas, Texas USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

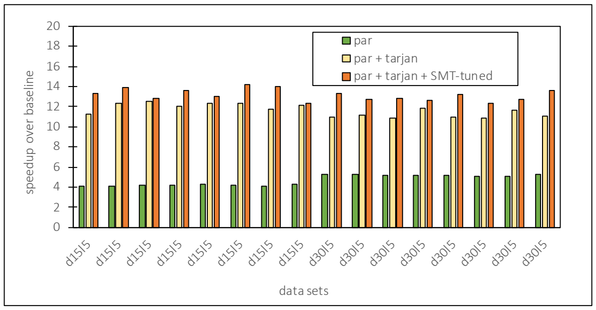

Figure 1: Performance improvements

labeled nodes to find the minimum cost. This is because there is no interest in identifying the nodes that are closer to the initial node v_0 in increasing order of cost. This feature can be exploited to make FTA more efficient in practice. The improvements do not affect the complexity of the algorithm, which remains at $(|E| + |V| log|V|)$.

4.1 Extracting Parallelism

Parallelism is extracted along the the following fronts:

Material Handling Cost: MHC calculations for each feasible layout are independent. We parallelize this computational task across the |V| layouts. Since |V| is large, we sub-divide the layout space and assign a thread to tackle a whole segment rather than a single layout. In the calculation of MHC for a single layout, a parallel reduction is performed across flows between each pair of facilities.

Relocation Cost: RC calculation between any two distinct pairs of layouts is independent. We parallelize the RC calculation across layouts for a given period.

Optimal path: We employ the FTA to find the most cost-effective sequence of configurations over the periods. The algorithm cannot be parallelized across periods due to dependencies. Nonetheless, appraising the weights for all edges emanating from a node in t to all nodes in $t + 1$ can be done concurrently. We partitioned the graph based on configurations that belong to different periods and threads explore configurations within each period in parallel.

Time Skewing: We observe that RC and MHC calculations for a given period does not interfere with the calculations of these cost in other periods. We take advantage of this property to parallelize RC and MHC computation along the time dimension. To improve locality and curb bandwidth pressure we further apply a skewing optimization over the time-parallelized loop.

Other: Construction of search space is also parallelized.

5 EXPERIMENTAL RESULTS

We evaluated our solution on the Stampede2 compute cluster at the TACC and on an IBM Power8 system. Stampede2 compute nodes host an Intel Xeon Skylake processor with 48 cores on two sockets. Each core is two-way hyper-threaded. The processor operates at 2.1 GHz. The Power8 system consists of 160 logical cores with 4-way NUMA partitioning. Each processor runs at 2.1 GHz. The problems sets were taken from [1].

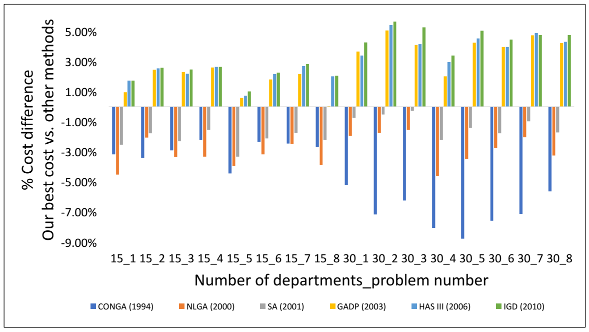

Figure 2: Solution accuracy vs. previous approaches

Fig. 1 shows performance improvements from our strategies over a sequential implementation The parallelization strategy (i.e. *par*) yields higher performance benefits for larger problem sizes, yielding as much as a 50% increased performance for problems with 30 facilities over the one with 6 facilities. However, the most significant performance gains come from enhancements to the FTA (i.e. par+tarjan). On average, the enhanced FTA doubles the performance The SMT-tuning method (i.e. par+tarjan+SMT-tuned) further improves the performance by close to 20%, resulting in an average of a factor of 13 speedup across all data sets.

Fig. 2 compares our best solutions to six previous methods. On average our solution is 3.46% of the best known solution (BKS) and in the worst case it is within 5.66% of the BKS.

6 CONCLUSIONS

Our parallel approach provides vast improvements in computational time over serial DFLP solutions and its best solutions are competitive if compared to existing methods using sophisticated meta-heuristics such as simulated annealing and genetic algorithms.

7 ACKNOWLEDGMENTS

This work was supported by NSF grants CNS-1253292 and OAC-1829644 and an equipment donation by IBM. The authors also acknowledge the Texas Advanced Computing Center (TACC) for the use of their computing resources.

REFERENCES

[1] J. Balakrishnan and C. H. Cheng. 200. Genetic search and the dynamic layout problem. *Computers and Operations Rsearch* (200), 587–593.
[2] J. Balakrishnan, F. R. Jacobs, and M. A. 1992. Solutions for the constrained dyanamic facility layout problem. *European Journal of Operational Rsearch* 57 (1992), 280–286.
[3] M. L. Fredman and R.E. Tarjan. 1987. Fibonacci Heaps and their uses in improved network optimization algorithms. *Journal of the association for computing machinery* 34, 3 (1987), 596–615.
[4] G.Moslemipour, T. S. Lee, and D. Rilling. 2012. A review of intelligent approaches for designing dynamic and robust layouts in flexible manufacturing systems. *International Journal of Advanced Manufacturing Technology* 60 (2012), 11–27.
[5] T. A. Lacksonen and E. Emory Enscore, Jr. 1993. Quadratic assignment algorithms for the dynamic layout problem. *International Journal of Production Research* 31, 3 (1993), 503–517.
[6] A. R. McKendall Jr. and J. Shang. 2006. Hybrid ant systems for the dynamic facility layout problem. *Computers and Operations Research* 33 (2006), 790–803.
[7] N. Nahas, M. Nourelfath, and D. AitKadi. 2010. *Iterated great deluge for the dynamic facility layout problem.* Technical Report. Centre interuniversitaire de reserche sur les reseaux dentreprise la logistique et le transport.
[8] M. J. Rosenblatt. 1986. The Dynamics of Plant Layout. *Management Science* 32 (1986), 76–86.