Proceedings of Proceedings of the ASME 2017 Dynamic Systems and Control Conference
DSCC2017
October 11-13, 2017, Tysons, Virginia, USA

DETC2009/MESA-12345

OPTIMAL REAL TIME RECONFIGURATION OF MANUFACTURING SYSTEMS BY SOFTWARE DEFINED CONVEYOR NETWORKS

Alireza Alizadegan*
Mechanical Engineering Department
University of Michigan
Ann Arbor, Michigan, 48109
aralizad@umich.edu

Kira Barton
Mechanical Engineering Department
University of Michigan
Ann Arbor, Michigan, 48109
bartonkl@umich.edu

Dawn Tilbury
Mechanical Engineering Department
University of Michigan
Ann Arbor, Michigan, 48109
tilbury@umich.edu

ABSTRACT

This paper introduces a method, named software defined conveyor networks (SDCN), for optimal real time reconfiguration of manufacturing systems by global view. SDCN is developed by extension of a method developed for cyber networks, named software defined networks (SDN), such that mechanical constraints present in manufacturing systems are considered. SDCN is formulated mathematically to prove optimality, scalability, and practical implementation by efficient algorithms in real time. The topology of a reconfigurable test bed is used as case study to illustrate effectiveness of SDCN by simulations.

INTRODUCTION

CONVEYOR NETWORKS PROBLEM

SDCN concerns conveyor networks defined as set of key points connected by conveyors where parts get transferred from one key point to another key point through conveyors. Conveyor is defined in this paper as follows:

Definition 1 (conveyor). *any unidirectional material handling device that transfers parts from one key point to another*

Following assumptions are made about conveyors:

C1: conveyors transfer parts continuously (e.g., no pallet stoppers)
C2: conveyor direction of motion does not change (e.g., no reversible conveyors)

Based on assumption C1, SDCN is concerned with transfer rate of parts in conveyor. Maximum transfer rate of parts in conveyor is referred to as capacity of conveyor.

Key points are defined as follows:

Definition 2 (key point). *any physical region in manufacturing system where at least one of following may occur:*

1. *parts are available to be transferred (e.g., raw material, output of process, etc.)*
2. *parts get handed off to different conveyors (e.g., distribution between or merge from multiple conveyors etc.)*
3. *parts exit from system (e.g., production of finished product, transfer of part to another facility)*

Following assumptions are made about key points:

K1: 1, 2, and 3 in definition 2 occur continuously and parts cannot stay in key points (e.g., no buffers or timed processes)
K2: all available parts in a key point have certain destination key points associated with them

Based on assumption K1, SDCN is concerned with rate with which 1, 2, and 3 occur. Available rate of parts to be transferred from certain source key point to certain destination key point (K2) is referred to as demand from source key point to destination.

Following assumptions are made about demands:

*Address all correspondence to this author.

D1: all demands in conveyor network are similar in terms of capacity they occupy in conveyor and priorities (e.g., value, deadline, etc.)

D2: demands are only concerned about source and destination key points and there are no concerns about intermediate key points they visit (e.g., requirements, precedence)

Based on definitions of key point and conveyor, path is defined as follows:

Definition 3 (path). *any sequence of key points in which each key point is connected to following key point by a conveyor in its direction of motion and no key point appears more than once.*

Based on assumptions C1 and K1, parts being transferred in a path must exit in the last key point of the path. Therefore, exit rate in a key point equals sum of transfer rate in all paths ending in that key point. As such sum of exit rate in all key points of a conveyor network, referred to as output rate, equals to sum of transfer rates in all paths in conveyor network; however, transfer rate of paths are constrained by capacity of conveyors that constitute paths and available transfer rate (demand). More specifically, sum of transfer rate in all paths that pass through a conveyor should be less than or equal to capacity of the conveyor for all conveyors. This constraint is referred to as capacity constraint (CC). Moreover, sum of transfer rate of paths with common first and last key points should not exceed demand from first key point to last key point. This constraint is referred to as demand constraint (DC). Therefore, output rate maximization problem is posed as follows: given conveyor network, capacities, and demands, determine transfer rates of all paths in conveyor network such that sum of them is maximized and capacity and demand constraints are satisfied.

SOFTWARE DEFINED CONVEYOR NETWORK METHOD

Terminologies and notations in this paper are mostly adopted from [1]. For set of positive real numbers, $\mathbb{R}_+$ is used. For an integer n, $[n]$ denotes the set $\{1,2,...,n\}$. For finite set X, cardinality of X (i.e., number of elements in X) is denoted by $|X|$, i.e., $X = \{x_1,...,x_{|X|}\}$. The Cartesian product of sets X_1, X_2,... , X_p is represented by $X_1 \times X_2 \times ... \times X_p = \{(x_1,x_2,...,x_p) : x_i \in x_i, \forall i \in [p]\}$. For function $f : X \rightarrow \mathbb{R}_+$, $[f(x_i)]_{i=1}^{|X|}$ denotes $|X| \times 1$ vector whose i^{th} element is $f(x_i)$. Similarly, for finite set $X = \{x_1,...,x_{|X|}\}$ and $Y = \{y_1,...,y_{|Y|}\}$ with function $f : X \times Y \rightarrow \mathbb{R}_+$, $[f(x_i,y_j)]_{i,j=1,1}^{|X|,|Y|}$ denotes $|X| \times |Y|$ matrix whose i, j^{th} element is $f(x_i,y_j)$. Vectorization of a matrix is a linear transformation which converts matrix into column vector. Specifically, vectorization of an $m \times n$ matrix A, denoted $vec(A)$, is vector of length mn obtained by stacking columns of matrix A on top of one another as:

$$\text{vec}(A) = [a_{1,1}, \ldots, a_{m,1}, a_{1,2}, \ldots, a_{m,2}, \ldots, a_{1,n}, \ldots, a_{m,n}]^{\mathrm{T}} \quad (1)$$

Function $ind2sub$ denotes transformation from indexes of elements in vectorized matrix to its subscripts in original form of matrix. For index $\zeta < mn$, this function performs transformation as $(i, j) = ind2sub(\zeta)$, where i and j are quotient and remainder of division ζ/m, respectively.

SDCN problem is solved based on SDN [2]; however, to account for unidirectional conveyors, directed graphs are used to represent conveyor networks, while undirected graphs are used to represent cyber networks in SDN. Graph representing conveyor network is denoted by $G(N,A)$, where $N = \{1,2,...,|N|\}$ denotes nodes of G representing key points and $A = \{1,2,...,|A|\}$ denotes arrows of G describing conveyors.

Adjacency and incidence matrices representing structure of graph of N and A are denoted by adj and inc, respectively. Demand is given as $|N| \times |N|$ matrix denoted by T. $T(n,n')$ represents demand from node n to node n'. Capacities are given as vector of length $|A|$ denoted by C each component of which represents capacity of corresponding arrow. Based on, adj and inc, set of paths in conveyor network is defined by formalizing definition 3 as:

$$\begin{aligned} P = \{p_i = (n_1,...,n_L) : n_l \in N,\ \forall l \in [L] \\ \wedge\ \text{adj}(n_l,n_{l+1}) = 1,\ \forall l \in [L-1] \\ \wedge\ n_i \neq n_j \text{ if } i \neq j,\ \forall i, j \in [L]\} \end{aligned} \quad (2)$$

where i denotes enumeration of path p_i and L denotes number of nodes of path in path p_i. The set of nodes in the path p_i is denoted by $N(p_i) = \{n_l : l \in [L]\}$. According to definition 3, each node in the path is connected to following node by an arrow in its direction. Therefore, set of arrows in the path p_i can be determined as $A(p_i) = \{a_l,\ l \in [L-1] : inc(n_l,a_l) = 1\ \wedge\ inc(n_{l+1},a_l) = -1\}$. A $(n,n\prime)$-path refers to a path from (n) to $(n\prime)$. Set of $(n,n\prime)$-paths is defined as $P_{n,n\prime} = \{p \in P : n_1 = n \wedge n_L = n\prime\}$. Members of set P and their enumerations can be obtained by analysis of graph by search algorithms (see Appendix). Transfer rate of each path is denoted by $t : P \rightarrow \mathbb{R}_+$.

Based on notations presented in this section, output rate maximization problem described in previous section can be formulated as:

$$\underset{t(p),\ \forall p\in P}{\text{maximize}} \sum_{p\in P} t(p) \qquad (*)$$

subject to

$$\sum_{p:a\in A(p)} t(p) \le c(a) \qquad \forall a\in A\ (CC)$$

$$\sum_{p\in P_{n,n\prime}} t(p) \le T(n,n') \qquad \forall n,n'\in N\ (DC)$$

$$t(p)\ge 0 \qquad \forall p\in P$$

In this optimization problem objective function is output rate which equals sum of transfer rate in all paths as explained in previous section. Optimization variables are transfer rate in paths for all paths in set of paths P. There are three sets of constraints. The first set correspond to capacity constraints. In these constraints, for each arrow a, constraint function makes sure that sum of transfer rate in all paths that the arrow belongs to arrow set of those paths is less than or equal to capacity of the arrow. The second set corresponds to demand constraints. In these constraints, for each ordered pair of nodes (n,n'), constraint function makes sure that sum of transfer rate in all $(n,n\prime)$-path is less than or equal to demand from n to n'.

First and second constraint functions represents CC and DC, while third constraint function ensures transfer rate of paths are positive quantities.

Proposition 1. *problem (*) can be formulated as standard linear programming (LP) problem*

Proof. consider standard LP problem of form:

$$\underset{x}{\text{minimize}}\ f^T x$$

subject to

$$\boldsymbol{A}x \le 0$$

$$x \ge 0$$

Optimization variables in (*) can be represented by definition of optimization vector of LP as stack of transfer rate of all paths in (2) as $x := [t(p_i)]_{i=1}^{|P|}$. Based on this definition of optimization vector, objective function of (*) can be represented by $f^T x$ if f is defined as $f := -1_{|P|\times 1}$, where minus signs account for change of problem from maximization to minimization.

To represent constraint functions, define function $f_c : A\times P \to \{0,1\}$ that determines whether an arrow is in a path or not as follows:

$$f_c(a,p) = \begin{cases} 1 & a\in A(p) \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

Based on this function, define matrix $\boldsymbol{A_c} := [f_c(a,p_i)]_{a,i=1,1}^{|A|,|P|}$. Using function f_c, this matrix's (a,i) element is 1 if $a\in A(p)$ and 0 otherwise. By defining vector $b_c := C$, inequality $A_c x \le b_c$ equals capacity constraints (CC). Based on (1), vectorized form of demand matrix $T(n,n')$ is denoted by $vec(T)$ and indexes of pair (n,n') in $vec(T)$ is denoted by ζ. Based on this, define function $f_d : N\times N\times P \to \{0,1\}$ that determines whether a path is associated with ordered pair of nodes corresponding to ζ as follows:

$$f_d(\zeta,p) = \begin{cases} 1 & ind2sub(\zeta) = (n_1,n_L) \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

Based on this function, define matrix $\boldsymbol{A_d} := [f_d(\zeta,p_i)]_{\zeta,i=1,1}^{|N|^2,|P|}$. Using function f_d, this matrix's (ζ,i) element is 1 if $p_i\in P_{n,n'}$ and 0 otherwise. By defining vector $b_d := vec(T)$, inequality $A_d x \le b_d$ equals demand constraints (DC). Both $A_c x\le b_c$ and $A_d x \le b_d$ are represented simultaneously by inequality constraint of LP ($Ax <= b$) through definition of its parameters as $\boldsymbol{A} = [\boldsymbol{A_c}^T \boldsymbol{A_d}^T]^T$ and $b = [b_c^T b_d^T]^T$. Finally, based on proposed definition of x constraints for positivity of value of transfer rate of paths is equivalent to $x\ge 0$.

□

Remarks

Formulation of optimization problem as linear program suggests that there exists efficient algorithms [3] to solve the SDCN problem in real time, solution of problem is global optimal, and SDCN method can be applied to large scale and complex conveyor networks. The proof could also be used as practical steps of implementation of proposed method.

According to formulation, changes in T and C merely needs LP to be solved again; however, changes in conveyor network requires execution of path search algorithm as well as evaluation of $\boldsymbol{A_c}$ and $\boldsymbol{A_d}$ which are not necessarily as efficient as LP. Therefore, main use of proposed method is suggested for reconfiguration in response to demand variations.

SIMULATIONS

APPENDIX: DEPTH FIRST SEARCH ALGORITHM FOR DIRECTED GRAPHS

This section presents details of algorithms used to finding and enumerating paths in conveyor networks. The problem is equivalent to finding all paths in cyclic directed graph which can be solved by modified depth first search algorithm [4]. Details of employed algorithm is presented in Algorithm 1 and 2. Algorithm 1 performs depth first search for all nodes by employing recursive procedure presented in Algorithm 2. As shown in line 6 of Algorithm 2, direction of conveyors are considered by adjacency matrix during finding paths.

Algorithm 1 Modified depth first search algorithm for finding and enumerting all simple paths in a cyclic directed graph

Inputs: N, adj
Output: p_i, $\forall i$

$i = 0$
for $n \in N$ **do**
  $l = \emptyset$ ▷ l denotes list of current sequence of nodes
  $v = 0_{|N|\times 1}$ ▷ v denotes whether node is visited
  DFS(n) ▷ Recursive procedure presented in Algorithm 2
end for
return p_i, $\forall i$

Algorithm 2 Recursive depth first search procedure

1: **procedure** DFS(n)
2:   $v(n) \leftarrow 1$
3:   $l' \leftarrow l$ ▷ l' denotes old list of sequence of nodes
4:   $l \leftarrow (l, n)$
5:   **for** $n' \in N$ **do**
6:     **if** $adj(n, n') = 1 \wedge v(n') = 0$ **then**
7:       $i \leftarrow i + 1$
8:       $p_i \leftarrow (l, n')$
9:       DFS(n')
10:     **end if**
11:   **end for**
12:   $v(n) \leftarrow 0$
13:   $l \leftarrow l'$
14:   **return** P
15: **end procedure**

APPENDIX: FORMALISM FOR OBTAINING DISTRIBUTION AT DISTRIBUTION NODES FROM TRANSFER RATES IN PATHS

This section describes calculation distribution of transfer rates in distribution nodes from transfer rates in paths. To this goal, first distribution nodes are obtained as:

$$D = \{d \in N : \sum_{n=1}^{|N|} adj(d,n) > 1\} \tag{5}$$

For each distribution node d obtained from equation 5, set of option nodes are defined as:

$$O = \{o \in N : adj(d,o) = 1\} \tag{6}$$

Distribution law of distribution node d is represented by function $dist : O \times [|N|^2] \rightarrow \mathbb{R}_+$. Distribution law represents fraction of transfer rate of parts from node n to node n' that distribution node d should send to option o.

$$dist(o,(n,n')) = \{\frac{\sum_{p\in H} t(p)}{\sum_{p\in I} t(p)}\} \tag{7}$$

where sets H and I are defined as follows:

$$H = \{p \in P_{n,n'} : \exists a \in A(p), inc(d,a) = 1, inc(o,a) = -1\} \tag{8}$$

$$I = \{p \in P_{n,n'} : d \in N(p)\} \tag{9}$$

Set H includes (n,n') paths that go through arrow connecting distribution node to its option node, while set I includes (n,n') paths that go through distribution node.

ACKNOWLEDGMENT

REFERENCES

[1] Bang-Jensen, J., and Gutin, G. Z., 2008. *Digraphs: theory, algorithms and applications*. Springer Science & Business Media.
[2] Hu, Y., Wang, W., Gong, X., Que, X., Ma, Y., and Cheng, S., 2015. "Maximizing network utilization in hybrid software-defined networks". In Global Communications Conference (GLOBECOM), 2015 IEEE, IEEE, pp. 1–6.
[3] Luenberger, D. G., 1973. *Introduction to linear and nonlinear programming*, Vol. 28. Addison-Wesley Reading, MA.
[4] Pieterse, V., and Black, P. E. Dictionary of Algorithms and Data Structures.