

Special Issue On: "Unbalanced Production Line Systems: Merits and Applications"

International Journal of Manufacturing Technology and Management

Guest Editor: Dr. Sabry Shaaban

METRICS AND EXPERIMENTAL DATA FOR ASSESSING UNBALANCED DISASSEMBLY
LINES

Seamus M. McGovern, Electronics Engineer and Volpe Fellow, U.S. DOT National Transportation Systems Center, 55 Broadway, Cambridge, Massachusetts 02142-1093 USA; Phone: (617)-494-2054, Fax: (617)-494-3666, E-mail: seamus.mcgovern@dot.gov

Surendra M. Gupta*, Professor and Director of the Laboratory for Responsible Manufacturing, 334 Snell Engineering Center, Department of Mechanical and Industrial Engineering, Northeastern University, 360 Huntington Avenue, Boston, Massachusetts 02115-5000 USA; Phone: (617)-373-4846, Fax: (617)-373-2921, E-mail: gupta@neu.edu
(*Corresponding author)

Biographical notes:

Seamus McGovern is an Electronics Engineer and Volpe Fellow at the U.S. DOT National Transportation Systems Center. He holds a commission in the Navy Reserve as an Aerospace Engineering Duty Officer as well as a part-time faculty appointment at Northeastern University. A former Helicopter Aircraft Commander, Instructor Pilot, and Maintenance Test Pilot, he holds an M.S. in Systems and Control Engineering, a Ph.D. in Industrial Engineering, and is a credentialed Project Management Professional. A member of the Society of Flight Test Engineers and reviewer for five journals, he has been elected to three scholastic honor societies, co-authored more than three-dozen technical papers and book chapters, and is the recipient of four fellowships, eleven merit-based scholarships, and a national best dissertation prize.

Dr. Surendra M. Gupta is a Professor of Industrial Engineering and Director of Laboratory for Responsible Manufacturing at Northeastern University in Boston. He received his BE in Electronics Engineering from Birla Institute of Technology and Science, MBA from Bryant University, and MSIE and PhD in Industrial Engineering from Purdue University. He has authored and co-authored well over 375 technical papers published in prestigious journals, books and conference proceedings. He has traveled to all seven continents and presented his work at international conferences on six of them. He is the recipient of Outstanding Research and Outstanding Industrial Engineering Professor awards from Northeastern University. His recent activities can be viewed at <<http://www1.coe.neu.edu/~smgupta/>>.

Metrics and experimental data for assessing unbalanced disassembly lines

Abstract: Disassembly lines are inherently multi-criteria, with balance having the possibility of being one of the lower priorities. This is due to the fact that other criteria – for example, removing valuable or hazardous materials early on in the process – may be of more importance than obtaining an optimal balance. In addition, complete disassembly may not be desired, required, or even possible, resulting in only partial disassembly being conducted. The result may be a disassembly sequence that readily satisfies the decision maker’s primary requirements, but at the expense of an unbalanced line. However, obtaining the decision maker’s primary requirements may not sufficiently justify exceptionally poor balance. Alternatively, the difference between a balanced disassembly line and one that is unbalanced – but at the expense of other desirable criteria – may turn out to be so insignificant that a focus purely on balance may obfuscate the benefits of considering other criteria. Therefore, metrics are also needed to quantitatively evaluate the merits of all considered criteria, including the level of balance (or unbalance). In this paper, a multi-criteria benchmark data set and associated metrics are developed for use in quantitatively evaluating an unbalanced paced disassembly line. These data and metrics are then demonstrated using an established disassembly line model and a deterministic, multi-criteria search heuristic.

Keywords: paced production system, disassembly, multi-criteria decision making, benchmark data set, metrics, reverse manufacturing, modeling, unbalanced lines

1. PROBLEM INTRODUCTION

Unbalancing an un-paced production line has been shown to potentially increase production rates when task times are stochastic and appropriate buffers are provided between workstations (Hillier and Boling, 1966 and 1979). Unbalancing can also be an unintended consequence of disassembly lines, even when considering a paced line and deterministic task times.

Disassembly lines are the result of manufacturers increasingly recycling and remanufacturing their post-consumed products as a result of new, more rigid environmental legislation, increased public awareness, and extended manufacturer responsibility. The economic attractiveness of reusing products, subassemblies, or parts instead of disposing of them has further fueled this effort. *Recycling* is a process performed to retrieve the material content of used and non-functioning products. *Remanufacturing*, on the other hand, is an industrial process in which worn-out products are restored to like-new conditions. Thus, remanufacturing provides the quality standards of new products with used parts. The first crucial step of both of these processes is disassembly.

Disassembly is defined as the methodical extraction of valuable parts/subassemblies and materials from discarded products through a series of operations. After disassembly, reusable parts/subassemblies are cleaned, refurbished, tested, and directed to inventory for remanufacturing operations. The recyclable materials can be sold to raw-material suppliers, while any residuals are sent to landfills. The multi-objective Disassembly Line Balancing Problem (DLBP) seeks to find a disassembly solution schedule which: provides a feasible disassembly sequence, minimizes the number of workstations, minimizes total idle time, and ensures similar idle times at each workstation (as well as addressing other, disassembly-specific concerns).

In this paper, the multi-objective Disassembly Line Balancing Problem is defined and metrics are demonstrated in order to measure various aspects of the performance of a given heuristic's ability to generate a desirable part removal sequence. A varying size, known optimal solution, benchmark data set is then developed and analyzed. As DLBP has been shown to be NP-complete (McGovern and Gupta, 2007), heuristics are generally needed to provide a part removal sequence. Due to sub-optimal nature of heuristics, especially when considering larger problem instances, the ability to quantitatively measure a given heuristic's ability to satisfy multiple disassembly criteria on a paced line while attempting to balance the line is of interest. Here, a general-purpose deterministic search algorithm is used as a case study heuristic to demonstrate how the various disassembly criteria under consideration perform using various sizes of the benchmark data set. These results are then used to evaluate the heuristic using the developed measures and metrics.

2. LITERATURE REVIEW

A major part of manufacturing and assembly operations, the assembly line is a production line where material moves continuously at a uniform rate through a sequence of workstations where assembly work is performed. With research papers going back to the 1950s, the Assembly Line Balancing problem is well defined and fairly well understood. While having differences from assembly line balancing, the recent development of DLBP requires that related problems be fully investigated and understood in order to better define DLBP and to obtain guidance in the search for appropriate methodologies to model and to solve it. Gutjahr and Nemhauser (1964) first described a solution to the Assembly Line Balancing problem, while Erel and Gokcen (1999) developed a modified version by allowing for mixed-model lines (assembly lines used to assemble different models of the same product). Suresh *et al.* (1996) first presented a genetic algorithm to provide a near-optimal solution to the Assembly Line Balancing problem. Tabu search is used in balancing assembly lines in Lapierre *et al.* (2006) on the Simple Assembly Line Balancing problem type I (SALB-I) with instances from the literature (*Arcus 1* and *2*) and a case study from industry. Hackman *et al.* (1989) proposed a branch-and-bound heuristic for the SALB-I problem. Ponnambalam *et al.* (1999) compared line-balancing heuristics using a quantitative evaluation of six assembly line balancing techniques.

Unbalancing a production line was first proposed by Hillier and Boling (1966) in recognition of studies conducted modeling flow shops using queueing theory. Modeling was accomplished with independent queues in series, with an infinite queue before the first server and a finite buffer of capacity s for all others. Each workstation was modeled individually as an $M/E_k/1/s+1$ queue ($M/E_k/1$ for the first workstation, where M represents arrivals according to a Poisson distribution and E_k represents service according to an Erlang distribution with shape parameter k) and having probabilistic *arrival rates* (in agreement with the nature of an un-paced line; i.e., a flow shop with buffers) and probabilistic *service rates* (conceptually equivalent to probabilistic task times). It was found, not only that balancing the line was detrimental to its production rate, but that unbalancing should be performed in such a way that the largest amounts of work should be allocated at the end workstations and the least amounts allocated to the ones in the middle (Hillier and Boling, 1966 and 1979, and Stecke and Solberg, 1985).

Several recent papers have discussed the different aspects of end-of-life processing. Brennan *et al.* (1994) and Gupta and Taleb (1994) investigated the problems associated with disassembly planning and scheduling. Torres *et al.* (2004) reported a study for non-destructive automatic disassembly of personal computers. Güngör and Gupta (1999b, 1999c, 2002)

presented the first introduction to disassembly line balancing and then developed an algorithm for solving the DLBP in the presence of failures with the goal of assigning tasks to workstations in a way that probabilistically minimizes the cost of defective parts (2001). McGovern *et al.* (2003) first proposed combinatorial optimization techniques for the DLBP. For a review of environmentally conscious manufacturing and product recovery see Güngör and Gupta (1999a). For a comprehensive review of disassembly sequencing see Lambert (2003) and Lambert and Gupta (2005).

3. DISASSEMBLY LINE MODEL

3.1. Introduction

The specific problem demonstrated in this paper seeks to fulfill five objectives:

1. minimize the number of disassembly workstations and hence, minimize the total idle time
2. ensure the idle times at each workstation are similar
3. remove hazardous components early in the disassembly sequence
4. remove high-demand components before low-demand components
5. minimize the number of direction changes required for disassembly

A major constraint is the requirement to provide a feasible disassembly sequence for the product being investigated. The result is an integer, deterministic, n -dimensional, multiple criteria decision-making problem with an exponentially growing search space (where n represents the number of parts for removal). Solutions consist of an ordered sequence (i.e., n -tuple) of elements; if a DLBP solution consisted of the eight-tuple $\langle 5, 2, 8, 1, 4, 7, 6, 3 \rangle$, then component 5 would be removed first, followed by component 2, then component 8, and so on.

While different authors use a variety of definitions for the term “balanced” in reference to assembly (Elsayed and Boucher, 1994) and disassembly lines, we apply the following definition (McGovern *et al.*, 2003; McGovern and Gupta, 2008) that considers the total number of workstations NWS and the station times ST_j (the total processing time requirement in workstation j); this definition will be used consistently throughout the paper:

Definition: A line is optimally balanced when the fewest possible number of workstations is needed and the variation in idle times between all workstations is minimized, while observing all constraints. This is mathematically described by

$$\text{Minimize } NWS$$

then

$$\text{Minimize } [\max (ST_x) - \min (ST_y)] \forall x, y \in \{1, 2, \dots, NWS\}$$

Line balancing can be visualized as in Figure 1. The five large boxes represent workstations with the total height of these boxes indicating cycle time CT (the maximum time available at each workstation). The smaller numbered boxes represent each part (1 through 11 here) with each being proportionate in height to its corresponding part removal time. The gray area is indicative of the idle time at each workstation.

>> *Insert Figure 1 about here* <<

3.2. *Model and measures*

A mathematical model is developed to quantitatively describe disassembly-line-related objective functions and performance measures.

Theorem: Let PRT_k be the part removal time for the k^{th} of n parts where CT is the maximum amount of time available to complete all tasks assigned to each workstation. Then for the most efficient distribution of tasks, the minimum number of workstations, NWS^* satisfies

$$NWS^* \geq \left\lceil \frac{\sum_{k=1}^n PRT_k}{CT} \right\rceil \quad (1)$$

Proof: If the above inequality is not satisfied, then there must be at least one workstation completing tasks requiring more than CT of time, which is a contradiction.

Subsequent bounds are shown to be true in a similar fashion and are presented throughout the paper without proof.

The upper bound for the number of workstations is given by

$$NWS_{nom} = n \quad (2)$$

therefore

$$\left\lceil \frac{\sum_{k=1}^n PRT_k}{CT} \right\rceil \leq NWS \leq n \quad (3)$$

Testing a given solution against the precedence constraints fulfills the major constraint of precedence preservation. Minimizing the total idle time I , which will also minimize the total number of workstations, attains objective 1 and is described by

$$I = (NWS \cdot CT) - \sum_{k=1}^n PRT_k \quad (4)$$

or

$$I = \sum_{j=1}^{NWS} (CT - ST_j) \quad (5)$$

This objective is represented as

$$\text{Minimize } Z_1 = \sum_{j=1}^{NWS} (CT - ST_j) \quad (6)$$

Line balancing seeks to achieve *perfect balance* (i.e., all idle times equal to zero). When this is not achievable, either Line Efficiency (LE) or the Smoothness Index (SI) is often used as a performance evaluation tool (Elsayed and Boucher, 1994). SI rewards similar idle times at each workstation, but at the expense of allowing for a large (sub-optimal) number of workstations. This is because SI compares workstation elapsed times to the largest ST_j instead of to CT . (SI is very similar in format to the sample standard deviation from the field of statistics, but using $\max(ST_j) \mid j \in \{1, 2, \dots, NWS\}$ rather than the mean of the station times.) LE rewards the minimum number of workstations but allows unlimited variance in idle times between workstations because no comparison is made between ST_j s. The balancing method used here (McGovern *et al.*, 2003, McGovern and Gupta, 2008) seeks to simultaneously minimize the number of workstations while ensuring that idle times at each workstation are similar, though at the expense of the generation of a nonlinear objective function. A resulting minimum numerical value is indicative of a more desirable solution, providing both a minimum number of workstations and similar idle times across all workstations. The measure of balance F is represented as

$$F = \sum_{j=1}^{NWS} (CT - ST_j)^2 \quad (7)$$

with the DLBP balancing objective represented as

$$\text{Minimize } Z_2 = \sum_{j=1}^{NWS} (CT - ST_j)^2 \quad (8)$$

Perfect balance is indicated by

$$Z_2 = 0 \quad (9)$$

Note that mathematically, Formula (8) effectively makes Formula (6) redundant due to the fact that it concurrently minimizes the number of workstations.

The lower bound on F is given by the optimal balance F^* where

$$F^* \geq \left(\frac{I}{NWS^*} \right)^2 \cdot NWS^* \quad (10)$$

while the upper bound is described by the worst case balance F_{nom} as

$$F_{nom} = \sum_{k=1}^n (CT - PRT_k)^2 \quad (11)$$

therefore

$$\left(\frac{I}{NWS^*}\right)^2 \cdot NWS^* \leq F \leq \sum_{k=1}^n (CT - PRT_k)^2 \quad (12)$$

Note that, in order to make the balance results comparable in magnitude to all subsequent metrics and to allow for more legible graphic comparisons with worst-case calculations in charts, the effects of squaring portions of Formula (7) can be normalized by taking the square root of the final balance measure calculated. For example, solutions having an equal number of workstations (e.g., $NWS = 3$) but differing idle times at each workstation (I_j) resulting in differing balance such as $I_j = \langle 1, 1, 4 \rangle$ and $I_j = \langle 2, 2, 2 \rangle$ (the latter is optimal) would have balance values of 18 and 12 respectively, while the normalized values would stand at 4.24 and 3.46, still indicating better balance with the latter solution, but also giving a sense of the relative improvement that solution provides, which the measure generated by Formula (7) lacks.

A hazard measure H was developed to quantify each solution sequence's performance in removing hazardous parts, with a lower calculated value being more desirable. This measure is based on binary variables that indicate whether a part is considered to contain hazardous material (set equal to one if the part is hazardous, else zero) and its position in the sequence. A given solution sequence hazard measure is defined as the sum of hazard binary flags multiplied by their position in the solution sequence, thereby rewarding the removal of hazardous parts early in the part removal sequence. This measure is represented as

$$H = \sum_{k=1}^n (k \cdot h_{PS_k}) \quad h_{PS_k} = \begin{cases} 1, & \text{hazardous} \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

where PS_k identifies the k^{th} part in the solution sequence PS ; i.e., for solution $\langle 3, 1, 2 \rangle$, $PS_2 = 1$. The DLBP hazardous part objective is represented as

$$\text{Minimize } Z_3 = \sum_{k=1}^n (k \cdot h_{PS_k}) \quad (14)$$

The lower bound on the hazardous part measure is given by

$$H^* = \sum_{p=1}^{|HP|} p \quad (15)$$

where the set of hazardous parts HP is defined as

$$HP = \{k : h_k \neq 0 \forall k \in P\} \quad (16)$$

and where P is the set of n part removal tasks. Its cardinality can be calculated with

$$|HP| = \sum_{k=1}^n h_k \quad (17)$$

For example, a product with three hazardous parts would give an H^* value of $1 + 2 + 3 = 6$. The upper bound on the hazardous part measure is given by

$$H_{nom} = \sum_{p=n-|HP|+1}^n p \quad (18)$$

or alternatively

$$H_{nom} = (n \cdot |HP|) - |HP| \quad (19)$$

For example, three hazardous parts in a product having a total of twenty would give an H_{nom} value of $18 + 19 + 20 = 57$ or equivalently, $H_{nom} = (20 \cdot 3) - 3 = 60 - 3 = 57$. Formulae (15), (18), and (19) are combined to give

$$\sum_{p=1}^{|HP|} p \leq H \leq \sum_{p=n-|HP|+1}^n p = (n \cdot |HP|) - |HP| \quad (20)$$

Also, a demand measure D was developed to quantify each solution sequence's performance in removing high-demand parts, with a lower calculated value being more desirable. This measure is based on positive integer values that indicate the quantity required of this part after it is removed – or zero if it is not desired – and its position in the sequence. Any given solution sequence demand measure is defined as the sum of the demand value multiplied by their position in the sequence, rewarding the removal of high demand parts early in the part removal sequence. This measure is represented as

$$D = \sum_{k=1}^n (k \cdot d_{PS_k}) \quad d_{PS_k} \in \mathbf{N}, \forall PS_k \quad (21)$$

where \mathbf{N} represents set of natural numbers; i.e., $\{0, 1, 2, \dots\}$. The DLBP part demand objective is represented as

$$\text{Minimize } Z_d = \sum_{k=1}^n (k \cdot d_{PS_k}) \quad (22)$$

The lower bound on the demand measure (D^*) is given by Formula (21) where

$$d_{PS_1} \geq d_{PS_2} \geq \dots \geq d_{PS_n} \quad (23)$$

For example, three parts with demands of 4, 5, and 6 respectively would give a best-case value of $(1 \cdot 6) + (2 \cdot 5) + (3 \cdot 4) = 28$. The upper bound on the demand measure (D_{nom}) is given by Formula (21) where

$$d_{PS_1} \leq d_{PS_2} \leq \dots \leq d_{PS_n} \quad (24)$$

For example, three parts with demands of 4, 5, and 6 respectively would give a worst-case value of $(1 \cdot 4) + (2 \cdot 5) + (3 \cdot 6) = 32$.

Finally, a direction measure R was developed to quantify each solution sequence's performance in removing parts with the same orientation together, with a lower calculated value indicating minimal direction changes and hence a more desirable solution. This measure is based on a count of the direction changes. Integer values represent each possible direction (typically $r \in \{+x, -x, +y, -y, +z, -z\}$; in this case $|r| = 6$). These directions are expressed as

$$r_{PS_k} = \begin{cases} +1, \text{direction} & +x \\ -1, \text{direction} & -x \\ +2, \text{direction} & +y \\ -2, \text{direction} & -y \\ +3, \text{direction} & +z \\ -3, \text{direction} & -z \end{cases} \quad (25)$$

and are easily expanded to other or different directions in a similar manner. The direction measure is represented as

$$R = \sum_{k=1}^{n-1} R_k \quad R_k = \begin{cases} 1, r_{PS_k} \neq r_{PS_{k+1}} \\ 0, \text{otherwise} \end{cases} \quad (26)$$

with the DLBP direction part objective represented as

$$\text{Minimize } Z_5 = \sum_{k=1}^{n-1} R_k \quad (27)$$

The lower bound on the direction measure is given by

$$R^* = |r| - 1 \quad (28)$$

For example, for a given product containing six parts that are installed/removed in directions $r_k = (-y, +x, -y, -y, +x, +x)$, the resulting best-case value would be $2 - 1 = 1$ (e.g., one possible R^* solution containing the optimal, single-change of product direction would be: $\langle -y, -y, -y, +x, +x, +x \rangle$). In the specific case where the number of unique direction changes is one less than the total number of parts n , the upper bound on the direction measure would be given by

$$R_{nom} = |r| \quad \text{where } |r| = n - 1 \quad (29)$$

Otherwise, the measure varies depending on the number of parts having a given removal direction and the total number of removal directions. It is bounded by

$$|r| \leq R_{nom} \leq n - 1 \quad \text{where } |r| < n - 1 \quad (30)$$

For example, six parts installed/removed in directions $r_k = (+x, +x, +x, -y, +x, +x)$ would give an R_{nom} value of 2 as given by the lower bound of Formula (30) with a possible solution

sequence of $\langle +x, +x, -y, +x, +x, +x \rangle$. Six parts installed/removed in directions $r_k = (-y, +x, -y, -y, +x, +x)$ would give an R_{nom} value of $6 - 1 = 5$ as given by the upper bound of Formula (30) with a solution sequence of $\langle -y, +x, -y, +x, -y, +x \rangle$ for example.

In the special case where each part has a unique removal direction, the measures for R^* and R_{nom} are equal and are given by

$$R^* = R_{nom} = n - 1 \quad \text{where } |r| = n \quad (31)$$

Note that the above optimal and nominal hazard, demand, and direction formulae are dependent upon favorable precedence constraints that will allow for generation of these optimal or nominal measures.

3.3. Models and measures as prototypes

It is important to note that the H , D , and R metrics are also intended as forming the three basic prototypes of any additional disassembly line evaluation criteria. These three different models are then the basis for developing differing or additional objectives.

The H metric is used as the prototype for any binary criteria; for example a part could be listed according to the categories “valuable” and “not valuable.”

The D metric is used as the prototype for any known value (integer or real) criteria; for example a part can be assigned a D -type metric which contains the part’s actual dollar value.

The R metric is used as the prototype for any adjacency or grouping criteria; for example a part could be categorized as “glass,” “metal,” or “plastic” if it were desirable to remove parts together in this form of grouping.

3.4. Metrics

The primary mathematical evaluation tool developed for quantitative analysis is shown in Formula (32) and is subsequently referred to as the *efficacy index EI*. The efficacy index is the ratio of the difference between a calculated measure x and its worst-case measure x_{nom} to the measure’s sample range (i.e., the difference between the best-case measure x^* and the worst-case measure as given by: $\max(X_y) - \min(X_z) \mid y, z \in \{1, 2, \dots, |X|\}$ from the area of statistical quality control). It is expressed as a percentage and described by (with the vertical lines in Formulae (32) through (33) representing absolute value versus cardinality as is seen elsewhere in this paper)

$$EI_x = \frac{100 \cdot |x_{nom} - x|}{|x_{nom} - x^*|} \quad (32)$$

This generates a value between 0 and 100%, indicating the percentage of optimum for any given measure and any given combinatorial optimization methodology being evaluated. For example, the efficacy index formula for balance would read

$$EI_F = \frac{100 \cdot |F_{nom} - F|}{|F_{nom} - F^*|}$$

For the study of multiple data sets, as is the case in this paper, probability theory presents us with the concept of a sample mean. The sample mean of a method's efficacy index can be calculated using

$$\overline{EI}_x = \left(\sum_{i=1}^y \frac{100 \cdot |x_{nom} - x_i|}{|x_{nom} - x^*|} \right) / y \quad (33)$$

where y is the sample size (the number of data sets). While Formula (32) provides individual data set size efficacy indices – especially useful in demonstrating worst and best case as well as trends with instance size – Formula (33) allows a single numerical value that provides a quantitative measure of the location of the data center in a sample.

3.5. Multi-criteria considerations

One of the ways in which the complexity of DLBP manifests itself is with multiple, often conflicting objectives. The field of multiple-criteria decision making provides a variety of means for addressing the selection of a solution where several objectives exist. The study in this paper addresses the multiple criteria aspects of disassembly using preemptive (lexicographic) goal programming. Since level of unbalance is the primary consideration in this paper, additional objectives are only considered subsequently; that is, the heuristic first seeks to select the best-performing measure of balance solution; equal balance solutions are then evaluated for hazardous part removal positions; equal balance and hazard measure solutions are evaluated for high-demand part removal positions; and equal balance, hazard measure, and high-demand part removal position solutions are evaluated for the number of direction changes. This priority ranking approach was selected over a weighting scheme for its simplicity, ease in re-ranking the priorities, ease in expanding or reducing the number of priorities, due to the fact that other weighting methods can be readily addressed at a later time, and primarily to enable unencumbered efficacy analysis of the sample heuristic and demonstrated data instances.

4. KNOWN OPTIMAL SOLUTION BENCHMARK DATA SET

4.1. Background

Any DLBP solution methodology needs to be applied to a collection of test cases to demonstrate its performance as well as to identify its limitations. In addition, new methodologies must first have their developed software thoroughly tested by undergoing a verification and validation process. *Verification* consists of providing a wide range of inputs to a module of the software to ensure proper operation of an individual software component, while *validation* determines whether or not the program as a whole provides a correct output for given input, necessitating (in the case of DLBP and similar problems) varying-size data sets having known optimal results.

Benchmark data sets are common for many NP-complete problems, such as *Nugent15/20/30*, *Elshafei19*, and *Krarup30* for application to the QUADRATIC ASSIGNMENT PROBLEM and *Oliver30* and *RY48P* for the TRAVELING SALESPERSON PROBLEM (see Lawler *et al.* 1985 for problem overviews). Unfortunately, because of their size and their design, most of these existing data sets have no known optimal answer and new solutions are not compared to the optimal solution, but rather the best solution to date.

In addition, since DLBP is a recent problem, very few instances exist to study the performance of different heuristic solutions. With no appropriate benchmark data sets, it was

desirable to develop a set of instances for the DLBP in order to evaluate DLBP heuristic performance. The known optimal solution benchmark line balance data set described here readily allows for determination of *efficacy* (a method's effectiveness in finding good solutions).

4.2. Formulation

This size-independent a priori benchmark data set was generated based on the following. Since, in general, solutions to larger and larger DLBP instances cannot be verified as optimal (due to the time complexity of exhaustive search), it is proposed that instances be generated in such a way as to always provide a known solution. This was done by using part times consisting exclusively of prime numbers, further selected to ensure that no combinations of these part removal times allowed for any equal summations (in order to reduce the number of possible optimal solutions). For example, part removal times of 1, 3, 5, and 7 and $CT = 16$ would have minimum idle time solutions of not only one 1, one 3, one 5, and one 7 at each workstation, but various additional combinations of these as well since $1 + 7 = 3 + 5 = \frac{1}{2} CT$. Subsequently, the chosen instances were made up of parts with removal times of 3, 5, 7, and 11 and $CT = 26$. As a result, the optimal balance for all subsequent instances would consist of a perfect balance of precedence-preserving combinations of 3, 5, 7, and 11 at each workstation with idle times of zero. The size of this *A Priori* data set is then constrained by

$$n = x \cdot |PRT| : x \in \mathbf{Z}^+ \quad (34)$$

where \mathbf{Z}^+ represents set of positive integers; i.e., $\{1, 2, \dots\}$.

To further complicate the data (i.e., provide a large, feasible search space), only one part was listed as hazardous and this was one of the parts with the largest part removal time (specifically, the last one listed in the initial data; see Table 1). In addition, one part (the last listed, second largest part removal time component) was listed as being demanded. This was done so that only the hazardous sequencing and the demand sequencing would be demonstrated, while providing a slight solution sequence disadvantage to any purely greedy methodology (since two parts with part removal times of 3 and 5 are needed along with the larger part removal time parts to reach F^* , assigning hazardous and demanded parts to those smaller-part-removal-time parts may allow some methodologies to artificially obtain the initial F^* single workstation sequence). From each part-removal-time size, the first listed part was selected to have a removal direction differing from the other parts with the same part removal time. This was done to demonstrate direction selection while requiring any solution-generating methodology to move these first parts of each part-removal-time size encountered to the end of the sequence (i.e., into the last workstation) in order to obtain the optimal direction value of $R^* = 1$ (i.e., if the solution technique being evaluated is able to successfully place the hazardous and demanded parts towards the front of the sequence). Also, there were no precedence constraints placed on the sequence, a deletion that further challenges any method's ability to attain an optimal solution (see McGovern and Gupta, 2008 for a practical case study that also incorporates precedence constraints).

>> *Insert Table 1 about here* <<

Known optimal results include $F^* = 0$, $H^* = 1$, $D^* = 2$, $R^* = 1$. While this paper makes use of data with $|PRT| = 4$ unique part removal times, in general for any n parts consisting of this type of data, the following can be calculated:

$$NWS^* = \frac{n}{|PRT|} \quad (35)$$

$$NWS_{nom} = n \quad (36)$$

$$I^* = 0 \quad (37)$$

$$I_{nom} = \frac{n \cdot CT \cdot (|PRT| - 1)}{|PRT|} \quad (38)$$

$$F^* = 0 \quad (39)$$

with F_{nom} given by Formula (11).

4.3. Generating the benchmark

Formulae have been developed to generate all data parameters, as well as for calculating optimal and nominal measures, for any size instance (as constrained by Formula (34)).

Hazard values and measure are given by

$$h_k = \begin{cases} 1, & k = n \\ 0, & otherwise \end{cases} \quad (40)$$

$$H^* = 1 \quad (41)$$

$$H_{nom} = n \quad (42)$$

with demand values and measure given by

$$d_k = \begin{cases} 1, & k = \frac{n \cdot (|PRT| - 1)}{|PRT|} \\ 0, & otherwise \end{cases} \quad (43)$$

$$D^* = \begin{cases} 2, & H = 1 \\ 1, & \textit{otherwise} \end{cases} \quad (44)$$

$$D_{nom} = \begin{cases} n-1, & H = n \\ n, & \textit{otherwise} \end{cases} \quad (45)$$

and part removal direction values and measure given by

$$r_k = \begin{cases} 1, & k = 1, \frac{n}{|PRT|} + 1, \frac{2n}{|PRT|} + 1, \dots, \frac{(|PRT| - 1) \cdot n}{|PRT|} + 1 \\ 0, & \textit{otherwise} \end{cases} \quad (46)$$

$$R^* = 1 \quad (47)$$

$$R_{nom} = \begin{cases} 0, & n = |PRT| \\ 2 \cdot |PRT| - 1, & n = 2 \cdot |PRT| \\ 2 \cdot |PRT|, & \textit{otherwise} \end{cases} \quad (48)$$

Since $|PRT| = 4$ in this paper, each part removal time is generated by

$$PRT_k = \begin{cases} 3, & 0 < k \leq \frac{n}{4} \\ 5, & \frac{n}{4} < k \leq \frac{n}{2} \\ 7, & \frac{n}{2} < k \leq \frac{3n}{4} \\ 11, & \frac{3n}{4} < k \leq n \end{cases} \quad (49)$$

While the demand values as generated by Formula (43) are the preferred representation (due to the fact that the resulting small numerical values make it easy to interpret demand efficacy since $D = k$), algorithms that allow incomplete disassembly may terminate after placing the single demanded part in the solution sequence. In this case, Formulae (43) – (45) may be modified to give

$$d_k = \begin{cases} 2, & k = \frac{n \cdot (|PRT| - 1)}{|PRT|} \\ 1, & \textit{otherwise} \end{cases} \quad (50)$$

$$D^* = \begin{cases} 2 + \sum_{p=1}^n p, & H = 1 \\ 1 + \sum_{p=1}^n p, & \textit{otherwise} \end{cases} \quad (51)$$

$$D_{nom} = \begin{cases} n - 1 + \sum_{p=1}^n p, & H = n \\ n + \sum_{p=1}^n p, & \textit{otherwise} \end{cases} \quad (52)$$

4.4. Analysis

It is noted that a data set containing parts with equal *PRT*s and no precedence constraints will have more than one optimal solution. To properly gauge the performance of any solution-generating technique on the DLBP *A Priori* data, the size of the optimal solution set needs to be quantified.

From probability theory we know that, for example, with $n = 12$ and $|PRT| = 4$, the size of the set of optimally balanced solutions $|F^*|$ when using the DLBP *A Priori* data could be calculated as $(12 \cdot 9 \cdot 6 \cdot 3) \cdot (8 \cdot 6 \cdot 4 \cdot 2) \cdot (4 \cdot 3 \cdot 2 \cdot 1) = 17,915,904$ from Table 2.

>> *Insert Table 2 about here* <<

Grouping these counts by workstation and reversing their ordering enables one to more easily recognize a pattern.

(1	2	3	4)
(2	4	6	8)
(3	6	9	12)

It can be seen that the first row can be generalized as $(1 \cdot 2 \cdot 3 \cdot \dots \cdot |PRT|)$, the second as $(2 \cdot 4 \cdot 6 \cdot \dots \cdot 2 \cdot |PRT|)$, and the third as $(3 \cdot 6 \cdot 9 \cdot \dots \cdot 3 \cdot |PRT|)$. Expanding in this way, the number of optimally balanced solutions can be written as

$$|F^*| = (1 \cdot 2 \cdot 3 \cdot \dots \cdot (1 \cdot |PRT|)) \cdot (2 \cdot 4 \cdot 6 \cdot \dots \cdot (2 \cdot |PRT|)) \cdot \dots \cdot \left(\frac{n}{|PRT|} \cdot \frac{2n}{|PRT|} \cdot \frac{3n}{|PRT|} \cdot \dots \cdot n \right)$$

This can be written as

$$|F^*| = \prod_{x=1}^{|PRT|} x \cdot \prod_{x=1}^{|PRT|} 2x \cdot \prod_{x=1}^{|PRT|} 3x \cdot \dots \cdot \prod_{x=1}^{|PRT|} \frac{n}{|PRT|} \cdot x$$

and finally as

$$|F^*| = \prod_{x=1}^{|PRT|} x^{\frac{n}{|PRT|}} \cdot \prod_{y=1}^{\frac{n}{|PRT|}} y$$

or

$$|F^*| = \prod_{x=1}^{|PRT|} \prod_{y=1}^{\frac{n}{|PRT|}} x^{\frac{n}{|PRT|}} \cdot y \quad (53)$$

Since $|PRT| = 4$ in this paper, Formula (53) becomes

$$|F^*| = \prod_{x=1}^4 \prod_{y=1}^{\frac{n}{4}} x^{\frac{n}{4}} \cdot y \quad (54)$$

In our example with $n = 12$ and $|PRT| = 4$, Formula (54) is solved as

$$|F^*| = \prod_{x=1}^4 \prod_{y=1}^{\frac{12}{4}} x^{\frac{12}{4}} \cdot y = \prod_{x=1}^4 \prod_{y=1}^3 x^3 \cdot y = \prod_{x=1}^4 x^3 \cdot (1 \cdot 2 \cdot 3)$$

or

$$|F^*| = 1 \cdot 1 \cdot 1 \cdot (1 \cdot 2 \cdot 3) \cdot 2 \cdot 2 \cdot 2 \cdot (1 \cdot 2 \cdot 3) \cdot 3 \cdot 3 \cdot 3 \cdot (1 \cdot 2 \cdot 3) \cdot 4 \cdot 4 \cdot 4 \cdot (1 \cdot 2 \cdot 3)$$

which, when rearranged, can be written as the more familiar

$$|F^*| = (12 \cdot 9 \cdot 6 \cdot 3) \cdot (8 \cdot 6 \cdot 4 \cdot 2) \cdot (4 \cdot 3 \cdot 2 \cdot 1) = 17,915,904$$

Even when all objectives are considered, there still exist multiple optimal solutions, again due to the use of a data set containing parts with equal PRT s and no precedence constraints. Using probability theory with the example having $n = 12$ and $|PRT| = 4$, it is known that the size of the set of solutions optimal in F , H , D , and R , $|F^* \cap H^* \cap D^* \cap R^*|$, when using the DLBP *A Priori* data can be calculated as $(1 \cdot 1 \cdot 6 \cdot 3) \cdot (4 \cdot 3 \cdot 2 \cdot 1) \cdot (4 \cdot 3 \cdot 2 \cdot 1) = 10,368$ from Table 3.

>> *Insert Table 3 about here* <<

Repeating the technique of grouping these counts by workstation and reversing their ordering again reveals a pattern.

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 3 & 6 & 1 & 1 \end{pmatrix}$$

The middle elements will always be the same as those given by Formula (53) but with two fewer sets (due to different first and last workstation elements). The first row is always $(1 \cdot 2 \cdot 3 \cdot \dots \cdot |PRT|)$ since the directional elements in the *A Priori* data should always be together at the end (the beginning in this case since we reversed the sequence for readability) of any optimal solution sequence. The last row (again, reversed) is always $((n/|PRT|) \cdot (2n/|PRT|) \cdot (3n/|PRT|) \cdot \dots \cdot (|PRT|-2) \cdot n/|PRT| \cdot 1 \cdot 1)$ since there is only one hazardous part (optimal element position $k = 1$) and only one demanded part (optimal element position $k = 2$). Combining these components, the number of fully optimal solutions can be written as

$$|F^* \cap H^* \cap D^* \cap R^*| = (1 \cdot 2 \cdot 3 \cdot \dots \cdot |PRT|) \cdot$$

$$\left((1 \cdot 2 \cdot 3 \cdot \dots \cdot (1 \cdot |PRT|)) \cdot (2 \cdot 4 \cdot 6 \cdot \dots \cdot (2 \cdot |PRT|)) \cdot \dots \cdot \left(1 \cdot \left(\frac{n}{|PRT|} - 2 \right) \cdot 2 \cdot \left(\frac{n}{|PRT|} - 2 \right) \cdot 3 \cdot \left(\frac{n}{|PRT|} - 2 \right) \cdot \dots \cdot |PRT| \cdot \left(\frac{n}{|PRT|} - 2 \right) \right) \right) \cdot$$

$$\left(\left(\frac{n}{|PRT|} \right) \cdot \left(\frac{2n}{|PRT|} \right) \cdot \left(\frac{3n}{|PRT|} \right) \cdot \dots \cdot \left(\frac{(|PRT|-2) \cdot n}{|PRT|} \right) \cdot 1 \cdot 1 \right)$$

By replacing the second term with a modified version of Formula (53) and simplifying the first and third terms, this can be written as

$$|F^* \cap H^* \cap D^* \cap R^*| = \left(\prod_{y=1}^{|PRT|} y \right) \cdot \left(\prod_{y=1}^{|PRT|} \prod_{z=1}^{\frac{n}{|PRT|}-2} y^{\frac{n}{|PRT|}-2} \cdot z \right) \cdot \left(\prod_{x=1}^{|PRT|-2} \frac{n}{|PRT|} \cdot x \right)$$

Expanding the second term gives

$$|F^* \cap H^* \cap D^* \cap R^*| = \left(\prod_{y=1}^{|PRT|} y \right) \cdot \left(\prod_{y=1}^{|PRT|} y^{\frac{n}{|PRT|}-2} \left(\prod_{z=1}^{\frac{n}{|PRT|}-2} z \right) \right) \cdot \left(\prod_{x=1}^{|PRT|-2} \frac{n}{|PRT|} \cdot x \right)$$

Combining the first and second terms results in

$$|F^* \cap H^* \cap D^* \cap R^*| = \left(\prod_{y=1}^{|PRT|} y^{\frac{n}{|PRT|}-1} \left(\prod_{z=1}^{\frac{n}{|PRT|}-2} z \right) \right) \cdot \left(\prod_{x=1}^{|PRT|-2} \frac{n}{|PRT|} \cdot x \right)$$

or

$$|F^* \cap H^* \cap D^* \cap R^*| = \prod_{x=1}^{|PRT|-2} \frac{nx}{|PRT|} \cdot \prod_{y=1}^{|PRT|} \prod_{z=1}^{\frac{n}{|PRT|}-2} y^{\frac{n}{|PRT|}-1} \cdot z$$

with a constraint that

$$\frac{n}{|PRT|} > 2$$

Alternatively, this can be written as

$$|F^* \cap H^* \cap D^* \cap R^*| = \prod_{x=1}^{|PRT|-2} \frac{nx}{|PRT|} \cdot \prod_{y=1}^{|PRT|} \prod_{z=1}^a y^{\frac{n}{|PRT|}-1} \cdot z \quad (55)$$

where

$$a = \begin{cases} \frac{n}{|PRT|} - 2, & \text{if } \frac{n}{|PRT|} > 2 \\ 1, & \text{otherwise} \end{cases}$$

Since $|PRT| = 4$ in this paper, Formula (55) becomes

$$|F^* \cap H^* \cap D^* \cap R^*| = \prod_{x=1}^2 \frac{nx}{4} \cdot \prod_{y=1}^4 \prod_{z=1}^{\frac{n}{4}-2} y^{\frac{n}{4}-1} \cdot z \quad (56)$$

In our example with $n = 12$ and $|PRT| = 4$, Formula (56) is solved as

$$|F^* \cap H^* \cap D^* \cap R^*| = \prod_{x=1}^2 3x \cdot \prod_{y=1}^4 \prod_{z=1}^1 y^2 \cdot z$$

or

$$|F^* \cap H^* \cap D^* \cap R^*| = \prod_{x=1}^2 3x \cdot \prod_{y=1}^4 y^2$$

giving

$$|F^* \cap H^* \cap D^* \cap R^*| = ((3 \cdot 1) \cdot (3 \cdot 2)) \cdot ((1 \cdot 1) \cdot (2 \cdot 2) \cdot (3 \cdot 3) \cdot (4 \cdot 4))$$

which, when rearranged, can be written as the more familiar

$$|F^* \cap H^* \cap D^* \cap R^*| = (1 \cdot 1 \cdot 6 \cdot 3) \cdot (4 \cdot 3 \cdot 2 \cdot 1) \cdot (4 \cdot 3 \cdot 2 \cdot 1) = 10,368$$

Although the sizes of both DLBP *A Priori* optimal solution sets are quite large in these examples, they are also significantly smaller than the search space of $n! = 479,001,600$. As shown in Table 4, the number of solutions that are optimal in balance alone goes from 100% of n at $n = 4$, to 22.9% at $n = 8$, and to less than 1% at $n = 16$; as n grows, this percentage gets closer and closer to zero. The number of solutions optimal in all objectives goes from less than 8.3% of n at $n = 4$, to 0.12% at $n = 8$, dropping to effectively 0% at $n = 16$; again, as n grows, the percentage of optimal solutions gets closer and closer to zero.

>> *Insert Table 4 about here* <<

4.5. *Other optimization priorities*

While this *A Priori* data set is effective at providing a quantitative evaluation of the level of unbalance (along with other objectives), if the primary consideration is not the measure of balance, the results may not provide an adequate measure. In this case, the *A Priori* data set can be further modified to gather additional information, while still providing an optimal level of balance and optimal or near-optimal (but always known) values for the other metrics. For example, if the primary consideration was the evaluation of a heuristic's ability to remove hazardous parts as early (or as late) as possible, the parts designated as hazardous would consist of the entire latter half of each group of parts having the same *A Priori* data set part removal time. Alternatively, if the primary objective was demand, each group of parts having the same *A Priori* data set part removal time would contain parts having increasingly larger demand values. Finally, if using direction as the primary criteria, parts could be initialized with alternating directions. With these designs, perfect balance could still be achieved while attaining optimal placement for any of the other criteria, but it should be noted that only one of these extensions should be selected at any time to simplify the calculation of optimal values for the remaining two criteria.

Other extensions include adding additional prototypes, changing the multi-criteria ordering of priorities, evaluation under the condition of incomplete or partial disassembly, or deleting some evaluation criteria (this is especially applicable to the case where the data set would be used in measuring the unbalance resulting from a heuristic applied to an assembly line study rather than a disassembly line). The data may also lend itself to *design for disassembly* efforts, which seek to preemptively mitigate future disassembly complexities.

Finally, while this benchmark data set has not been designed with stochastic part removal times in mind or for use with an un-paced (i.e., job shop) scheduling model, these can be addressed in several ways. It is recognized that the work-sampling operations required to define the standard deviation of stochastic task times on an assembly line can be excessively time consuming (Tiacci *et al.*, 2003). Tiacci *et al.* (2003) studied the significance of the task times' standard deviation on the performance of the system model in order to determine if it is worth collecting enough data to calculate the standard deviation. Ultimately, they proposed a methodology for performing assembly line balancing with a reduced set of data by using the means of the stochastic task times as the sole inputs, that is, without their standard deviation. The *A Priori* benchmark could also be used in this fashion by setting $E[x] = \mu_x = x$ and $\sigma_x^2 = 0 \forall x \in \{PRT_k\}$. This would be applicable in other cases, including those where use of probabilistic data is performed purely deterministically (e.g., by assigning the deterministic average-case or worst-case part removal time to the problem) or effectively deterministically (e.g., calculate the lowest probability of any candidate task exceeding the cycle time based upon each candidate's means and variances then assign the task with the best discrete mean value; Elsayed and

Boucher, 1994). Alternatively, the stochastic part removal time could be represented deterministically as being within some multiple of σ ; e.g., if a part's removal time were given by a normal distribution, this time could be represented deterministically as being within 1σ , 2σ , 3σ (68.27%, 95.45%, or 99.73% respectively) or some other tolerance of interest.

4.6. Demonstration set

The final configuration of the benchmark as used here was 19 instances with instance size equally distributed from $n = 8$ to $n = 80$ in increments of $|PRT| = 4$. The size and range of the instances is considered appropriate, with small ns tested – which decreases the NWS value and tends to exaggerate less than optimal performance – as well as large, which demonstrates time complexity growth and efficacy changes with n .

5. CASE STUDY HEURISTIC

A search approach that provides a very fast, near-optimal solution to combinatorial optimization problems is used as a case study to demonstrate the degree of unbalance as problem size grows, as well as its effect on the other criteria. This heuristic generates a feasible solution using a modified exhaustive search technique to provide data sampling of all solutions. While the case study makes use of balance as the highest priority in the interest of both demonstrating the data set and the increasing level of unbalance, other priority orderings can be selected in the interest of gaining a sense of change in balance level with instance size and in comparison to balance as being the highest priority.

5.1. Background

In many search applications in the physical world (e.g., antisubmarine warfare, search and rescue) exhaustive search is not possible due to time or sensor limitations. In these cases, it becomes practical to sample the search space and operate under the assumption that, for example, the highest point of land found during the conduct of a limited search is either is the highest point in a given search area or is reasonably near the highest point. The search technique (McGovern and Gupta, 2008) in this paper's case study works by sampling the exhaustive solution set; that is, it searches the solution space in a method similar to an exhaustive search but in a pattern that skips solutions (conceptually similar to the STEP functionality in a FOR loop as found in computer programming) to significantly minimize the search space (Figure 2; the shading indicates solutions visited, the border represents the search space).

>> *Insert Figure 2 about here* <<

The *skip size* ψ can be as small as $\psi = 1$ or as large as $\psi = n$ (where n is the number of parts). Note that $\psi = 1$ and $\psi = n$ are not used since $\psi = 1$ is equivalent to exhaustive search and $\psi = n$ generates a trivial solution (it returns only one solution, that being the data in the same sequence it is provided in).

5.2. Search process

As a modified exhaustive search allowing for solution sampling, it searches similarly to depth-first search, and allowing for skips in the search sequence. Due to H-K's deterministic nature, the data can be entered as-given (*forward*), in reverse order (*reverse*), or in any other order, all of which may affect the solutions visited. In the basic H-K, searching a permutation, and with for

example $\psi = 2$, the first element in the first solution would be 1, the next element position would first consider 1, but since 1 is already in the solution (in element position one), element position two would be incremented by one and 2 would be considered and be acceptable. This is repeated for all of the elements until the first solution (i.e., $\langle 1, 2, 3, \dots, n \rangle$) is generated. For the next solution visited, the rightmost element in the current solution that is able to be incremented by an amount equal to ψ without exceeding n would be incremented by ψ ; if the solution element's new value is equal to some previous element's value, the current element's value is incremented by one (again, as long as this does not result in exceeding n). Any remaining element positions to the right would be filled lexicographically (smallest to biggest, left to right) with the remaining values, if any. This process of generating a solution is continued to the left until the first element position is reached. Continuing with the example, the element under consideration would then be 1, which would be incremented by $\psi = 2$ and, therefore, 3 would be considered and inserted as the first element position value. Since element 1 is not yet in the sequence, it would be placed in the second solution position, part 2 in the third, etc., and the process continues. For example, with $\{1, 2, 3, 4\}$, instead of considering the $4! = 24$ possible permutations, only five are visited by H-K with $\psi = 2$ and using forward-only ordered data: $\langle 1, 2, 3, 4 \rangle$, $\langle 1, 4, 2, 3 \rangle$, $\langle 3, 1, 2, 4 \rangle$, $\langle 3, 1, 4, 2 \rangle$, and $\langle 3, 4, 1, 2 \rangle$. With $n = 5$, instead of considering the $5! = 120$ possible permutations, only 16 are considered at $\psi = 2$ and using forward-only data as demonstrated in Table 5.

>> *Insert Table 5 about here* <<

5.3. DLBP considerations

For DLBP and with a product undergoing complete disassembly, H-K can be applied to the permutation of the number of parts n with some slight modifications. Each iteration of the H-K-generated solution is considered for feasibility (due to precedence constraints). If it is feasible, H-K then looks at each element in the solution and places that element using the next-fit rule (from the Bin-Packing problem application; once a bin has no space for a given item attempted to be packed into it, that bin is never used again even though a later, smaller item may appear in the list and could fit in the bin, see Hu and Shing, 2002). If it does not fit, a new workstation is assigned and previous workstations are never again considered. (Although NF does not perform as well as First-Fit, Best-Fit, First-Fit-Decreasing, or Best-Fit-Decreasing when used in the general BIN-PACKING problem, it is the only one of these rules that will work with a DLBP solution sequence due to the existence of precedence constraints.) When all of the work elements have been assigned to a workstation, the process is complete and the balance, hazard, demand, and direction measures are calculated, and the process is repeated. The best of all of the inspected solution sequences is then saved as the problem solution.

6. MULTI-CRITERIA QUANTITATIVE ANALYSIS DEMONSTRATION

The computer program was written in ANSI C++ and run on a 1.6GHz PM x86-family workstation. Because H-K is deterministic and performs no preprocessing of the data, the analyses performed here were run with the benchmark data presented in forward and then reverse order. The two results were then averaged and the two search times were added. Also, all of the instance sizes were run several times, allowing ψ to vary from $n - 10$ to $n - 1$ (e.g., at an instance size of $n = 25$, skip would vary as $15 \leq \psi \leq 24$) with the best solution from these ten searches kept. For small data sets the software was set up so that it would not attempt any skip

size smaller than $\psi = 3$ to avoid exhaustive or near-exhaustive searches (which would result in unrealistically large search times on small data sets). All studies were run multiple times to determine an average of the computation times.

The H-K technique can be seen below (Table 6) on the DLBP *A Priori* data presented forward and reverse at $n = 12$ and $3 \leq \psi \leq 11$ (note that without the minimum allowed value of $\psi = 3$, skip values would otherwise include $2 \leq \psi \leq 11$). H-K was able to find a solution optimal in the number of workstations, balance, and hazard. The solution found came from the reverse set and consisted of $NWS^* = 3$, $F^* = 0$, $H^* = 1$, $D = 10$ (optimal value is $D^* = 2$), and $R = 2$ (optimal value is $R^* = 1$). The time to complete the forward and reverse searches averaged 1.56 seconds.

>> *Insert Table 6 about here* <<

Problem size was varied to demonstrate how solution performance changes with increases in n . On the full range of data ($n = \{8, 12, 16, \dots, 80\}$), H-K found solutions with NWS^* workstations up to $n = 12$, then solutions with $NWS^* + 1$ workstations through data set 11 ($n = 48$), after which it stabilized at $NWS^* + 2$ (Figure 3). From Formula (32) H-K's efficacy index in number of workstations started at an optimal $EI_{NWS} = 100\%$, dropped to a low of $EI_{NWS} = 92\%$, then continuously climbed through to $EI_{NWS} = 97\%$ with an efficacy index sample mean in number of workstations of $\overline{EI}_{NWS} = 96\%$.

>> *Insert Figure 3 about here* <<

Increases in unbalance are seen with increases in data set size. A detailed view of balance performance with problem size can be seen in Figure 4.

>> *Insert Figure 4 about here* <<

While increases in unbalance are seen with increases in n , as a percentage of the overall range from best case to worst case, the normalized balance measure tends to decrease (i.e., improve) with increases in the data set size (Figure 5). The normalized balance efficacy index dropped from a high of $EI_F = 100\%$ to a low of $EI_F = 85\%$ at data set 3 ($n = 16$) then slowly climbed to $EI_F = 92\%$ giving a sample mean of $\overline{EI}_F = 92\%$.

>> *Insert Figure 5 about here* <<

The hazardous part was regularly sub-optimally placed. Hazard part placement stayed relatively consistent with problem size (though effectively improving as compared to the worst case, as illustrated by Figure 6). These results are as expected since hazard performance is designed to be deferential to balance and affected only when a better hazard measure can be attained without adversely affecting balance. The hazard measure's efficacy index fluctuates between $EI_H = 57\%$ and $EI_H = 100\%$, giving a sample mean of $\overline{EI}_H = 90\%$.

>> *Insert Figure 6 about here* <<

The high-demand part was also sub-optimally placed, though at a higher rate than the hazardous part (Figure 7). Its efficacy index fluctuates between $EI_D = 7\%$ and $EI_D = 103\%$ (due

to better than optimal placement in position $k = 1$ at the expense of hazard placement). The resulting demand measure sample mean is $\overline{EI}_D = 49\%$.

>> *Insert Figure 7 about here* <<

With part removal direction structured as to be deferential to balance, hazard, and demand, it was seen to decrease in performance when compared to the best case and when compared to the worst case (Figure 8). Though the part removal direction efficacy gets as high as $EI_R = 86\%$, by data set 5 ($n = 24$) it has dropped to $EI_R = 0\%$ and never rises higher again than $EI_R = 43\%$, resulting in a sample mean of $\overline{EI}_R = 20\%$.

>> *Insert Figure 8 about here* <<

The progressively less than optimal heuristic performance using this data set is a desirable characteristic of the benchmark, as it indicates its ability to readily quantify unbalance and other performance measures. The DLBP *A Priori* benchmark data is especially designed to challenge the solution-finding ability of a variety of search methods in order to enable a thorough quantitative evaluation of these method's performances in different areas.

The DLBP *A Priori* benchmark is also appropriate for time complexity studies. With the case-study heuristic, a smaller ψ or the inclusion of precedence constraints will increasingly move the H-K method towards the optimal solution. As shown in Figure 9, the time complexity performance of H-K provides the tradeoff benefit with the technique's performance, demonstrating the moderate increase in time required with problem size that grows markedly slower than the exponential growth of exhaustive search. Exhaustive and third-order curves are shown for comparison.

>> *Insert Figure 9 about here* <<

Based on Figure 9 and Figure 10, a 2nd-order polynomial regression model was used to fit the H-K curve (assuming the data and structure as described in this study).

>> *Insert Figure 10 about here* <<

The function describing an algorithm's running time on a computer processor $T(n)$ was fitted using regression and found to be $T(n) = 0.0033n^2 - 0.0002n + 0.2893$ (with coefficients determined using a common spreadsheet software application). The small coefficients are indicative of a relatively slow runtime growth in instance size. The coefficient of determination is calculated to be 0.9974, indicating 99.74% of the total variation is explained by the calculated linear regression curve (the anomaly seen in the H-K curve in Figure 10 is due to the software rule set up to prevent exhaustive or near-exhaustive searches at small n). As seen in Figure 11 this regression provides a very accurate fit. With a growth of $0.0033n^2 - 0.0002n + 0.2893$, the average-case time complexity of H-K curve (with forward and reverse data, and skip sizes of $n - 10 \leq \psi \leq n - 1$) can then be described as $O(n^2)$ or polynomial complexity. The deterministic, single iteration nature of H-K also indicates that the process would be no faster than this so it could be expected that the time complexity lower bound is $\Omega(n^2)$ and, therefore, H-K appears to have an asymptotically tight bound of $\Theta(n^2)$ as configured in this paper.

>> *Insert Figure 11 about here* <<

7. CONCLUSIONS

With their multi-criteria nature, disassembly lines that are structured as flow shops can inherently possess a tendency to be unbalanced, even when the intent is to achieve a balanced line. This is due to other criteria (e.g., removing valuable or hazardous materials early in the disassembly process) having the potential to be of more importance than obtaining an optimal balance. The resulting unbalanced line may sacrifice optimal balance in the interest of a disassembly sequence that satisfies the decision maker's primary requirements. As such, a set of metrics to measure a line's performance, including the level of unbalance – when task sequences are determined by a given heuristic was described – as was a benchmark data set having a known optimal solution (including measure of balance) for all instance sizes. The application of these in evaluating a heuristic's ability to disassemble products of various sizes on a paced line while meeting other, potentially conflicting priorities was demonstrated using a deterministic search heuristic.

REFERENCES

1. Brennan, L., Gupta, S. M., and Taleb, K. N. (1994) 'Operations Planning Issues in an Assembly/Disassembly Environment', *International Journal of Operations and Production Management*, Vol. 14, No. 9, pp. 57-67.
2. Elsayed, E. A. and Boucher, T. O. (1994) *Analysis and Control of Production Systems*, Prentice Hall, Upper Saddle River, New Jersey.
3. Erel, E. and Gokcen, H. (1999) 'Shortest-Route Formulation of Mixed-Model Assembly Line Balancing Problem,' *European Journal of Operational Research*, Vol. 116, pp. 194-204.
4. Güngör, A. and Gupta, S. M. (1999a) 'A Systematic Solution Approach to the Disassembly Line Balancing Problem', *Proceedings of the 25th International Conference on Computers and Industrial Engineering*, New Orleans, Louisiana, pp. 70-73.
5. Güngör, A. and Gupta, S. M. (1999b) 'Disassembly Line Balancing', *Proceedings of the 1999 Annual Meeting of the Northeast Decision Sciences Institute*, Newport, Rhode Island, pp. 193-195.
6. Güngör, A. and Gupta, S. M. (1999c) 'Issues in Environmentally Conscious Manufacturing and Product Recovery: A Survey', *Computers and Industrial Engineering*, Vol. 36, No. 4, pp. 811-853.
7. Güngör, A. and Gupta, S. M. (2001) 'A Solution Approach to the Disassembly Line Problem in the Presence of Task Failures', *International Journal of Production Research*, Vol. 39, No. 7, pp. 1427-1467.
8. Güngör, A. and Gupta, S. M. (2002) 'Disassembly Line in Product Recovery', *International Journal of Production Research*, Vol. 40, No. 11, pp. 2569-2589.

9. Gupta, S. M. and Taleb, K. (1994) 'Scheduling Disassembly', *International Journal of Production Research*, Vol. 32, No. 8, pp. 1857-1866.
10. Gutjahr, A. L. and Nemhauser, G. L. (1964) 'An Algorithm for the Line Balancing Problem', *Management Science*, Vol. 11, No. 2, pp. 308-315.
11. Hackman, S. T., Magazine, M. J., and Wee, T. S. (1989) 'Fast, Effective Algorithms for Simple Assembly Line Balancing Problems', *Operations Research*, Vol. 37, No. 6, pp. 916-924.
12. Hillier, F. S. and Boling, R. W. (1966) 'The Effect of Some Design Factors on the Efficiency of Production Lines with Variable Operation Times', *Journal of Industrial Engineering*, Vol. 17, pp. 651-658.
13. Hillier, F. S. and Boling, R. W. (1979) 'On the Optimal Allocation of Work in Symmetrically Unbalanced Production Line Systems with Variable Operation Times', *Management Science*, Vol. 25, No. 8, pp. 721-728.
14. Hu, T. C. and Shing, M. T. (2002) *Combinatorial Algorithms*, Dover Publications, Mineola, New York
15. Lambert, A. D. J. (2003) 'Disassembly Sequencing: A Survey', *International Journal of Production Research*, Vol. 41, No. 16, pp. 3721-3759.
16. Lambert, A. J. D. and Gupta, S. M. (2005) *Disassembly Modeling for Assembly, Maintenance, Reuse, and Recycling*, CRC Press, Boca Raton, Florida.
17. Lapierre, S. D., Ruiz, A., and Soriano, P. (2006) 'Balancing Assembly Lines With Tabu Search', *European Journal of Operational Research*, Vol. 168, No. 3, pp. 826-837.
18. Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., and Shmoys, D. B. (1985) *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, John Wiley & Sons, New York, New York.
19. McGovern, S. M. and Gupta, S. M. (2007) 'Combinatorial Optimization Analysis of the Unary NP-Complete Disassembly Line Balancing Problem', *International Journal of Production Research*, Vol. 45, No. 18-19, pp. 4485-4511.
20. McGovern, S. M. and Gupta, S. M. (2008) 'Deterministic Search Algorithm for Sequencing and Scheduling Problems', *Meta-heuristics for Scheduling in Industrial and Manufacturing Applications, Studies in Computational Intelligence 128*, F. Xhafa and A. Abraham, eds., Springer, London, UK, pp. 105-124.
21. McGovern, S. M., Gupta, S. M., and Kamarthi, S. V. (2003) 'Solving Disassembly Sequence Planning Problems Using Combinatorial Optimization', *Proceedings of the 2003 Northeast Decision Sciences Institute Conference*, Providence, Rhode Island, pp. 178-180.
22. Ponnambalam, S. G., Aravindan, P., and Naidu, G. M. (1999) 'A Comparative Evaluation of Assembly Line Balancing Heuristics', *The International Journal of Advanced Manufacturing Technology*, Vol. 15, pp. 577-586.
23. Suresh, G., Vinod, V. V., and Sahu, S. (1996) 'A Genetic Algorithm for Assembly Line Balancing', *Production Planning and Control*, Vol. 7, No. 1, pp. 38-46.

24. Stecke, K. E. and Solberg, J. J. (1985) 'The Optimality of Unbalancing Both Workloads and Machine Group Sizes in Closed Queueing Networks of Multiserver Queues', *Operations Research*, Vol. 33, No. 4, pp. 882-910.
25. Tiacci, L., Saetta, S., and Martini, A. (2003) 'A Methodology to Reduce Data Collection in Lean Simulation Modeling for the Assembly Line Balancing Problem', *Proceedings of Summer Computer Simulation Conference 2003*, Montreal, Canada, pp. 841-846.
26. Torres, F., Gil, P., Puente, S. T., Pomares, J., and Aracil, R. (2004) 'Automatic PC Disassembly for Component Recovery', *International Journal of Advanced Manufacturing Technology*, Vol. 23, No. 1-2, pp. 39-46.

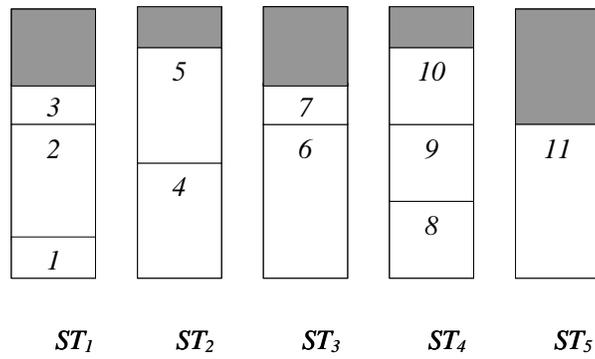


Figure 1 Line balancing depiction

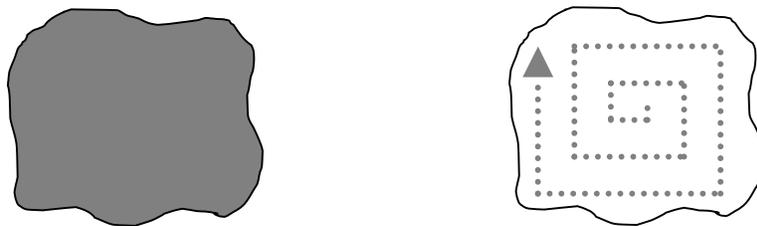


Figure 2 Exhaustive search space and the H-K search space and methodology

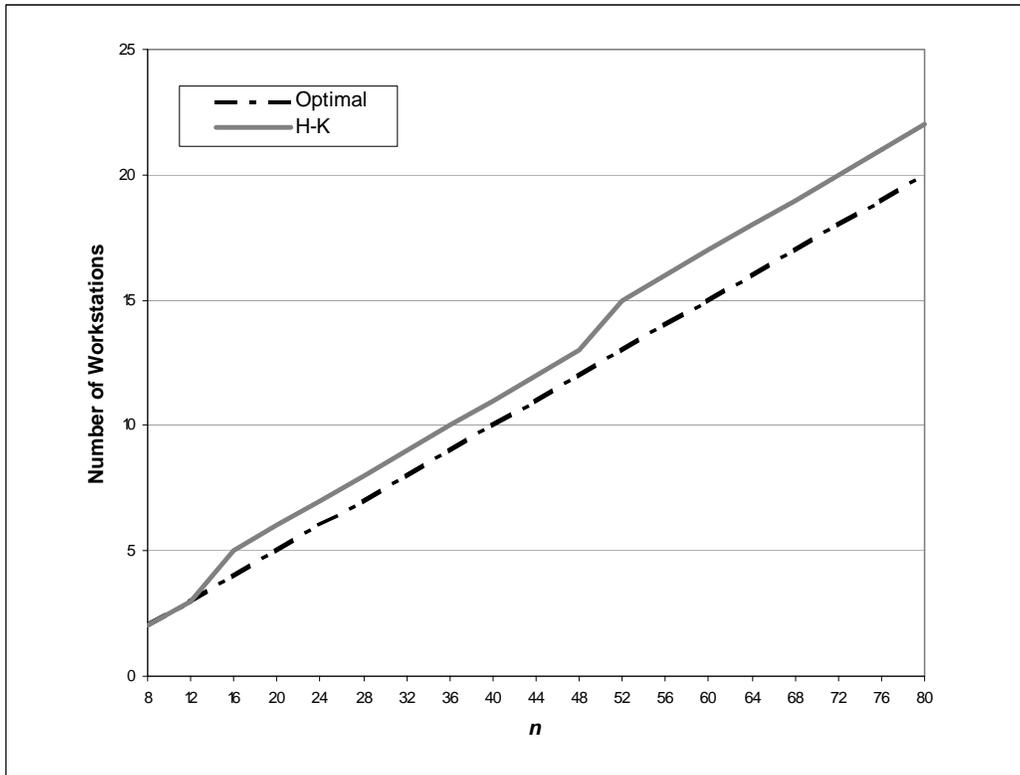


Figure 3 Workstation calculation

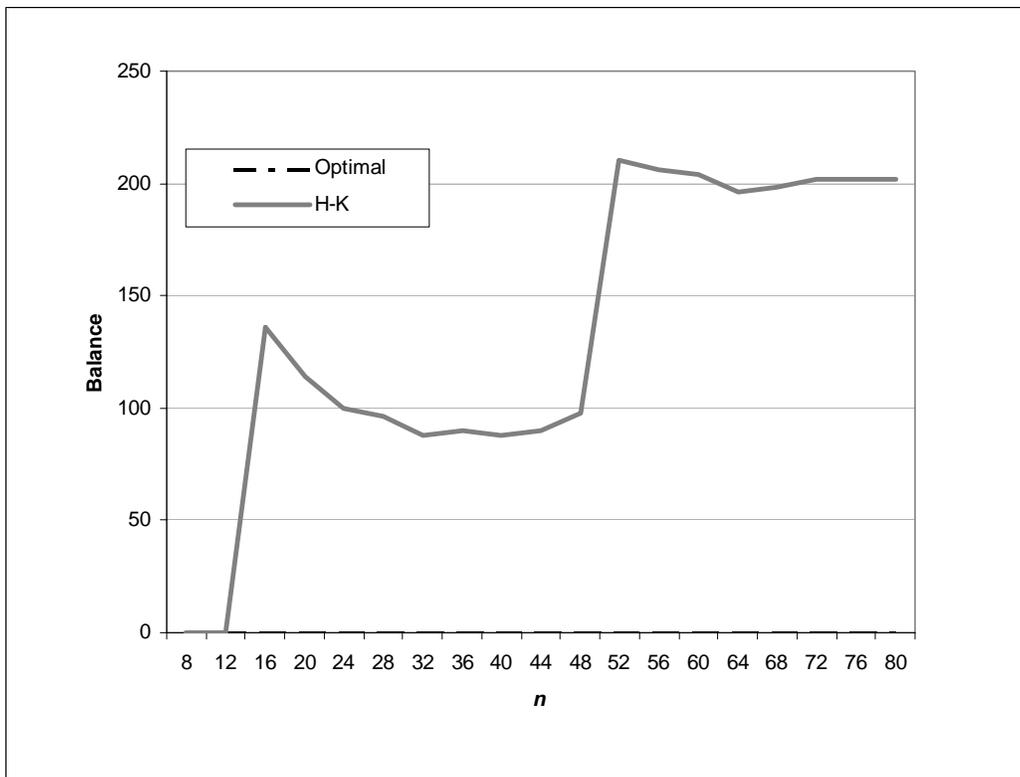


Figure 4 Detailed level of unbalance

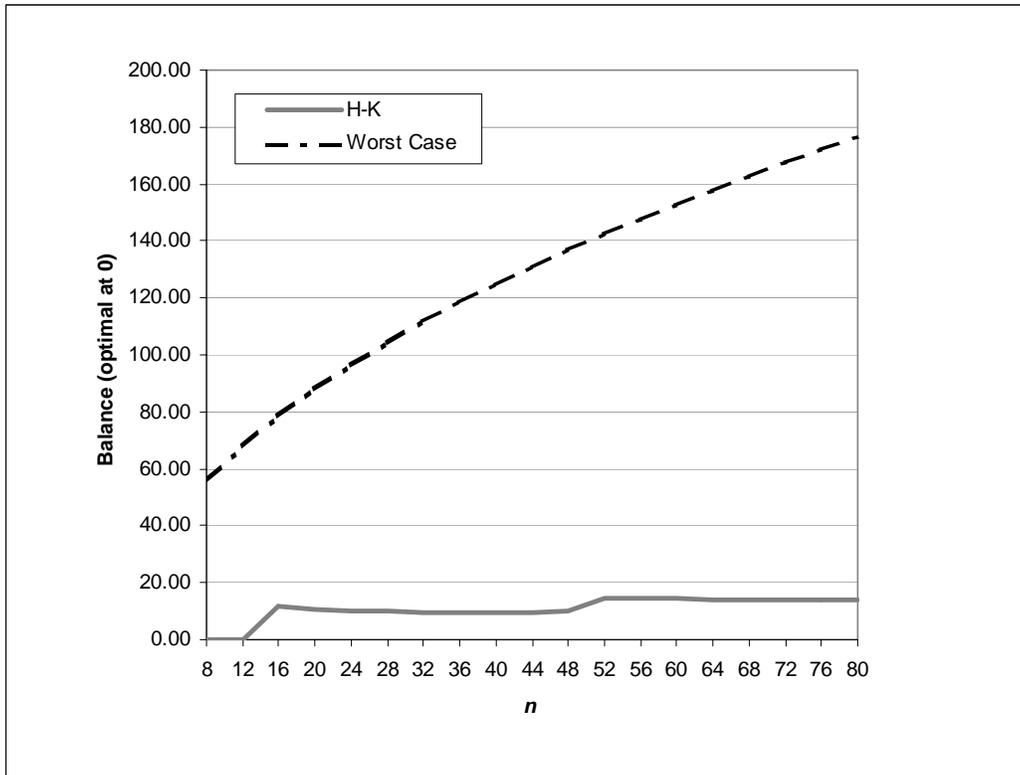


Figure 5 Normalized level of unbalance

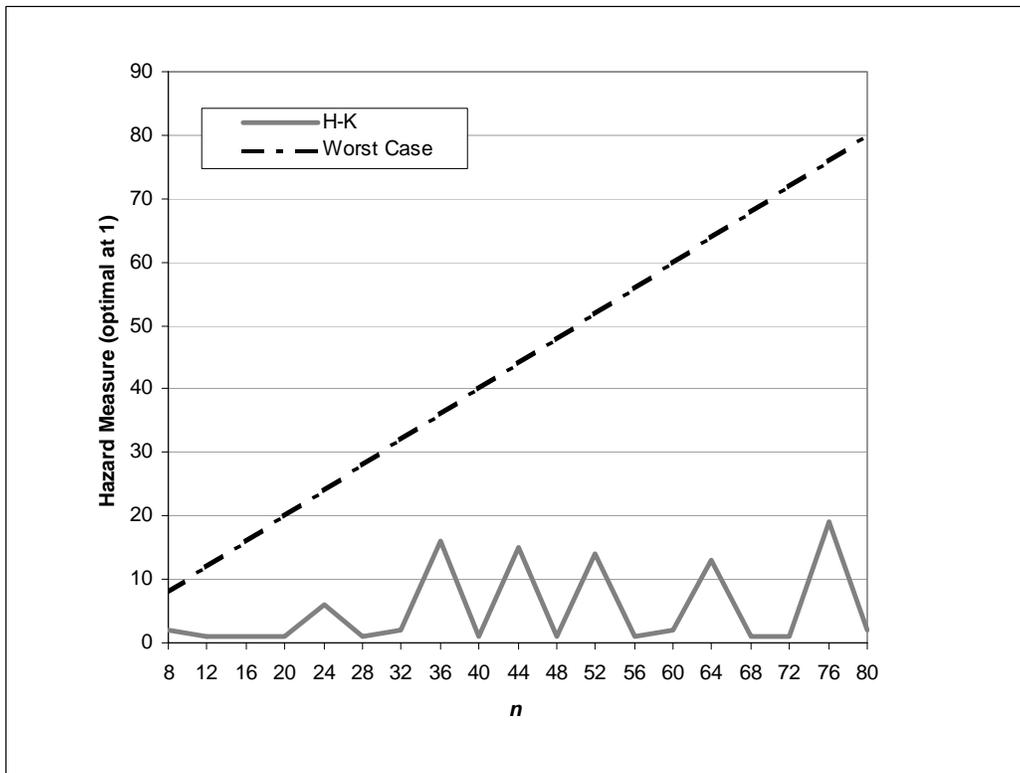


Figure 6 Hazard measure with instance size

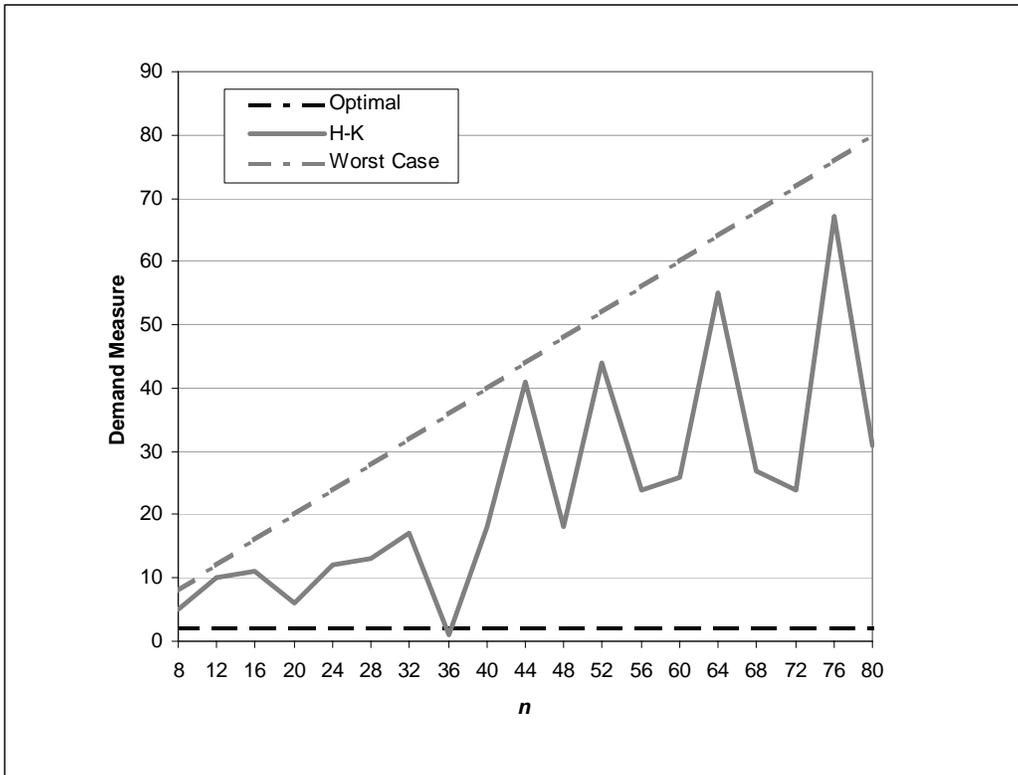


Figure 7 Demand measure with instance size

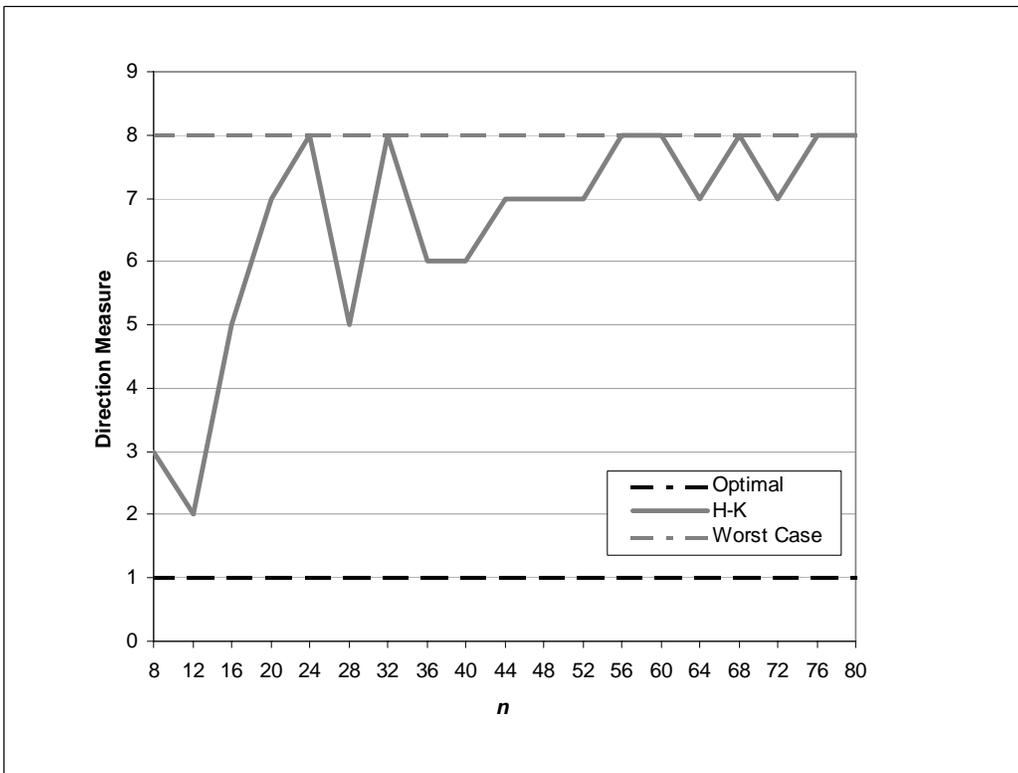


Figure 8 Part removal direction measure with instance size

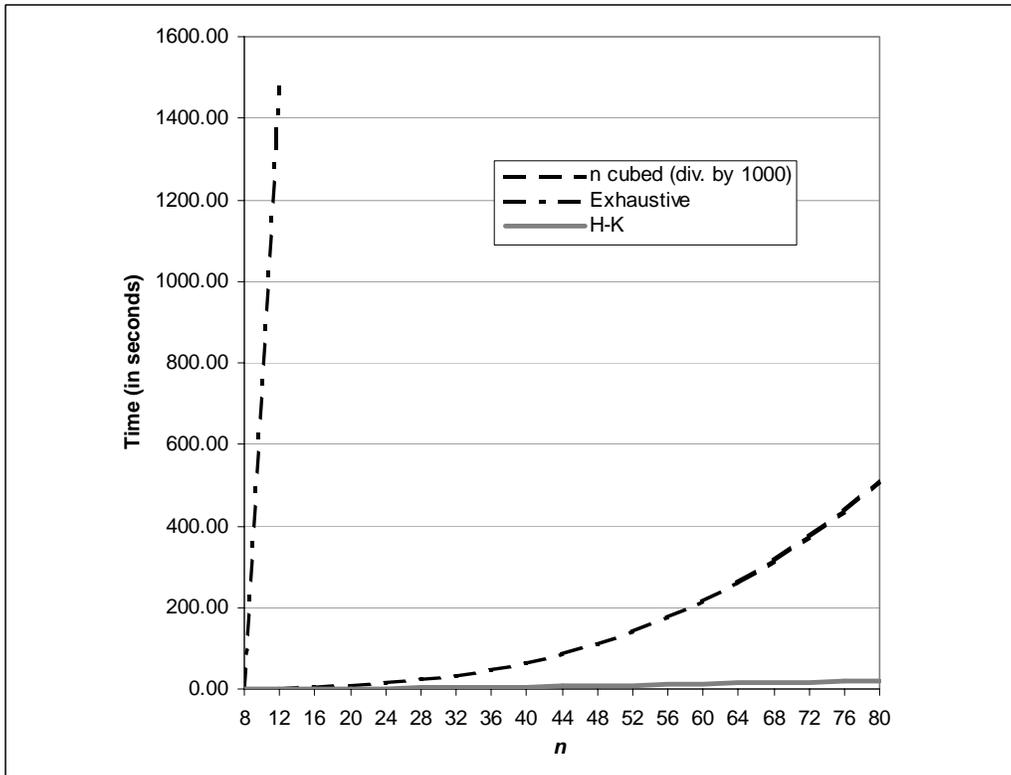


Figure 9 Heuristic's time complexity compared to exhaustive search

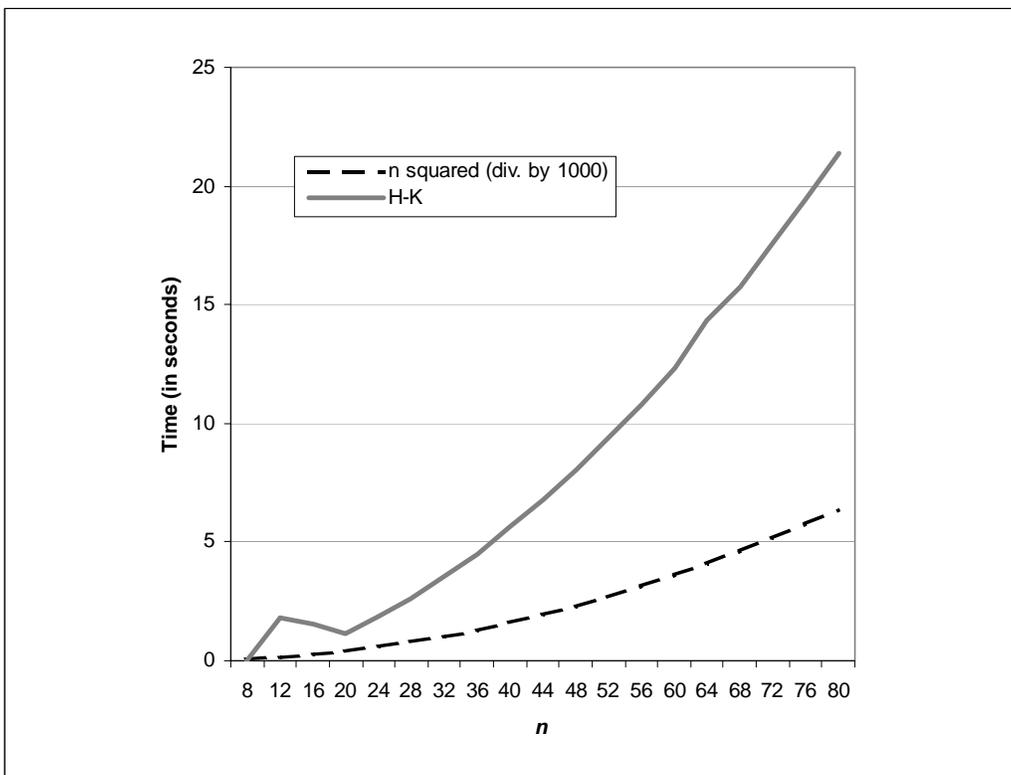


Figure 10 Heuristic's time complexity compared to second order

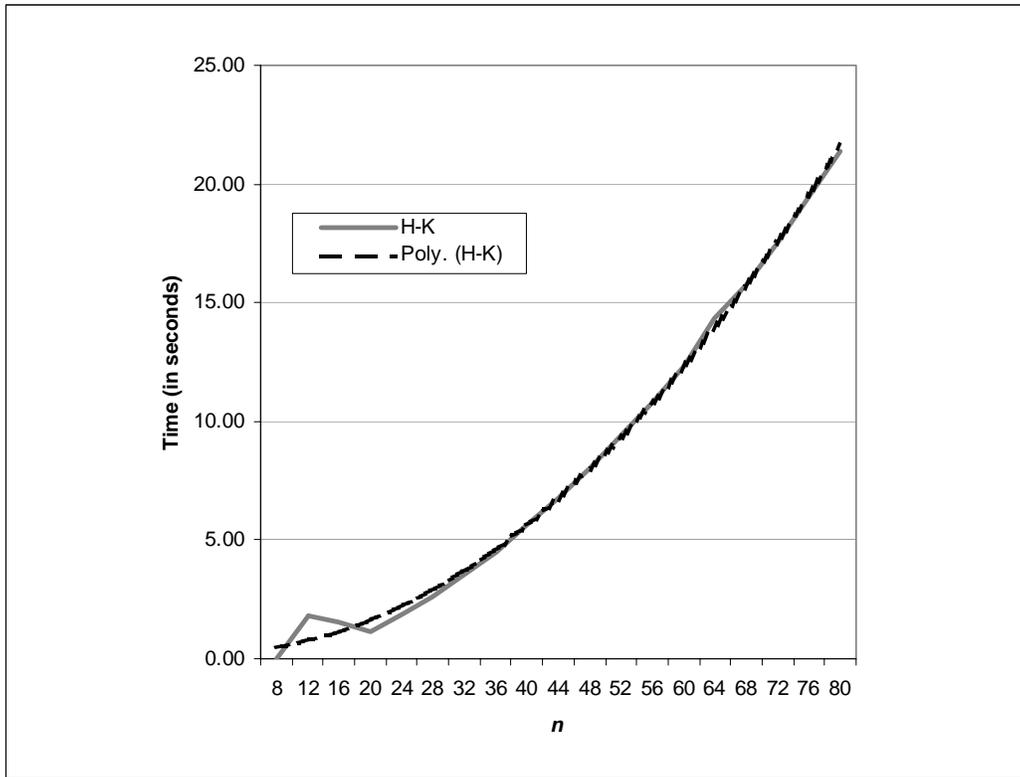


Figure 11 Detailed heuristic's time complexity and fitted second-order polynomial regression line

Table 1 DLBP *A Priori* data for $n = 12$

Part ID	1	2	3	4	5	6	7	8	9	10	11	12
<i>PRT</i>	3	3	3	5	5	5	7	7	7	11	11	11
Hazardous	0	0	0	0	0	0	0	0	0	0	0	1
Demand	0	0	0	0	0	0	0	0	1	0	0	0
Direction	1	0	0	1	0	0	1	0	0	1	0	0

Table 2 Number of possible entries in each element position resulting in perfect balance using the DLBP *A Priori* data with $n = 12$ and $|PRT| = 4$

k	1	2	3	4	5	6	7	8	9	10	11	12
count	12	9	6	3	8	6	4	2	4	3	2	1

Table 3 Number of possible entries in each element position resulting in optimal in F , H , D , and R using *A Priori* data with $n = 12$ and $|PRT| = 4$

k	1	2	3	4	5	6	7	8	9	10	11	12
count	1	1	6	3	4	3	2	1	4	3	2	1

Table 4 Comparison of possible solutions to optimal solutions for a given n using the DLBP *A Priori* data

n	$n!$	Number optimal in balance	Number optimal in all	Percentage optimal in balance	Percentage optimal in all
4	24	24	2	100.00%	8.33%
8	40,320	9,216	48	22.86%	0.12%
12	479,001,600	17,915,904	10,368	3.74%	0.00%
16	2.09228E+13	1.10075E+11	7,077,888	0.53%	0.00%

Table 5 H-K results at $n = 5$ and $\psi = 2$

$\langle 1, 2, 3, 4, 5 \rangle$	$\langle 3, 4, 1, 2, 5 \rangle$
$\langle 1, 2, 5, 3, 4 \rangle$	$\langle 3, 4, 1, 5, 2 \rangle$
$\langle 1, 4, 2, 3, 5 \rangle$	$\langle 3, 4, 5, 1, 2 \rangle$
$\langle 1, 4, 5, 2, 3 \rangle$	$\langle 5, 1, 2, 3, 4 \rangle$
$\langle 1, 4, 5, 3, 2 \rangle$	$\langle 5, 1, 4, 2, 3 \rangle$
$\langle 3, 1, 2, 4, 5 \rangle$	$\langle 5, 3, 1, 2, 4 \rangle$
$\langle 3, 1, 4, 2, 5 \rangle$	$\langle 5, 3, 1, 4, 2 \rangle$
$\langle 3, 1, 4, 5, 2 \rangle$	$\langle 5, 3, 4, 1, 2 \rangle$

Table 6 H-K solution using the *A Priori* instance at $n = 12$ and $3 \leq \psi \leq 11$ (forward and reverse)

Part ID	12	2	5	8	11	1	4	7	10	9	6	3
PRT	11	3	5	7	11	3	5	7	11	7	5	3
Workstation	1	1	1	1	2	2	2	2	3	3	3	3
Hazardous	1	0	0	0	0	0	0	0	0	0	0	0
Demand	0	0	0	0	0	0	0	0	0	1	0	0
Direction	0	0	0	0	0	1	1	1	1	0	0	0