*IIE Transactions* (2003) **35**, 605–617
Copyright © 2003 "IIE"
0740-817X/03 $12.00+.00
DOI: 10.1080/07408170390214473

# Using ant techniques to solve the assembly line balancing problem

PATRICK R. McMULLEN[1] and PETER TARASEWICH[2]

[1]*Babcock Graduate School of Management, Wake Forest University, Winston-Salem, NC 27106, USA*
*E-mail: pat.mcmullen@mba.wfu.edu*
[2]*College of Computer and Information Science, Northeastern University, 360 Huntington Avenue, 161CN, Boston, MA 02115, USA*
*E-mail: tarase@ccs.neu.edu*

Received July 2001 and accepted November 2002

This paper presents an approach, based on ant techniques, to effectively address the assembly line balancing problem with the complicating factors of parallel workstations, stochastic task durations, and mixed-models. A methodology was inspired by the behavior of social insects in an attempt to distribute tasks among workers so that strategic performance measures are optimized. This methodology is used to address several assembly line balancing problems from the literature. The assembly line layouts obtained from these solutions are used for simulated production runs so that output performance measures (such as cycle time performance) are obtained. Output performance measures resulting from this approach are compared to output performance measures obtained from several other heuristics, such as simulated annealing. A comparison shows that the ant approach is competitive with the other heuristic methods in terms of these performance measures.

## 1. Introduction

The Assembly Line Balancing (ALB) problem has received a great deal of attention over the years. In general, the problem seeks to schedule a given number of tasks among workstations (according to precedence relationships) so that a product can be completed within a given cycle time (time between successive units coming off of the assembly line). The ALB problem becomes more complicated when additional issues are addressed, such as stochastic task times, multiple workstation workers, and mixed-model assembly lines.

Even with today's ever-increasing computing power, the ALB problem, which is an NP-hard combinatorial optimization problem (McMullen and Frazier, 1998), is very difficult to solve to completion in a reasonable amount of time. Finding the optimal solution to these problems through enumeration of all possible combinations is usually impractical, but search heuristics can be used to find good solutions to the ALB problem in a reasonable amount of time. The typical objective associated with the ALB problem is to minimize the required number of workers, while at the same time making certain that there is ample capacity to complete all jobs within a given cycle time. Mathematically, this can be defined as follows:

$$\text{Min} \sum_{j=1}^{R} \dot{w}_j, \tag{1}$$

subject to

$$t_j \leq C\dot{w}_j, \tag{2}$$

where $R$ is the total number of work centers, $\dot{w}_j$ is the number of workers required in work center $j$, $C$ is the cycle time, and $t_j$ is the expected duration of all tasks in work center $j$.

The specific ALB problem studied in this research starts with a given number of products to be assembled, their demand, and a list of tasks required to produce each product. The expected duration of each task for each product is given, along with the standard deviation of each task. There is a set of precedence relationships given for the order in which the tasks must be completed for each product. The decision-maker chooses a desired cycle time, and a factor that affects the probability of opening a new work center. The solved problem results in a set of workstations, each with an assigned set of tasks.

In this paper, we present a heuristic that uses concepts derived from Ant Colony Optimization (ACO) techniques. ACO, part of a larger group of solution methods known as swarm intelligence, is a meta-heuristic that uses artificial ants to solve combinatorial optimization problems (Bonabeau *et al.*, 1999). There are two contributions made by this paper. First, the problem addressed is the mixed-model, stochastic, general ALB problem with the complicating feature of task paralleling. This problem is complex and difficult to solve, and has received little research attention to date. Second, and most significantly, the problem is solved using a heuristic that incorporates features based on ant techniques. The solution method developed exploits the unique properties of the ALB problem and produces good solutions in reasonable times using concepts borrowed from research in the area of ACO.

This paper is organized as follows. Section 2 reviews research on the ALB problem and introduces ACO techniques. Section 3 describes the ant-system related heuristic used to solve the ALB problem. Section 4 details a simulation study used to test the performance of the ant heuristic against other methods. Section 5 presents the results and an analysis of the simulation testing. Section 6 provides a conclusion and directions for future research.

## 2. Background

This section first provides a description of the ALB problem, along with some background on previous research efforts into solving the problem at hand and its variations. Next, an introduction to ACO techniques and current research in the area is given.

### 2.1. *ALB*

ALB is the practice of placing definable units of work (tasks) into groups, or work centers. When this type of problem became recognized in the 1950s, the problem's objective was faithful to its name, balancing the workload across all work centers. This practice of "balancing" was intended to maximize efficiency. Since then, however, researchers have pursued many different objectives. Some researchers still pursue ALB solutions that attempt to evenly distribute workloads across work centers, but others concern themselves with cost-minimization objectives, such as solutions encouraging on-time performance.

Ghosh and Gagnon (1989) have performed an extensive review of ALB research, and created a classification scheme for the ALB literature. The scheme first differentiates assembly lines as producing either a single model of a product or mixed models. These two categories are further broken down by whether the problem assumes deterministic or stochastic task times. A final step is classification by whether the problem addresses a simple or more general case. General cases can include factors such as multiple workstations, paralleling (allowing multiple workers at a single workstation), zoning restrictions and alternative flow policies (such as U-shaped layouts).

Given the comprehensiveness of Ghosh and Gagnon's work, the interested reader is directed to their paper for details on earlier ALB research. However, some of the more recent developments are summarized here according to the Ghosh and Gagnon framework. First looking at the single-model, deterministic problem, branch and bound methods continue to receive much attention (Hoffman, 1992; Sprecher, 1999). Easton *et al.* (1989) developed two network-based algorithms to solve the ALB problem. Anderson and Ferris (1994) and Leu *et al.* (1994) used genetic algorithm techniques. Malakooti (1991) implemented a multiple criteria decision-making approach, which was later expanded to include buffers (Malakooti, 1994).

Additional research has also been performed on solving the single-model, stochastic ALB problem. Sarin *et al.* (1999) used a procedure that combines dynamic programming with a branch and bound strategy. Lyu (1997) approached the problem using single-run optimization, which applies a stochastic optimization algorithm to a simulation model. Nkasu and Leung (1995) developed a stochastic ALB methodology, which combines a stochastic simulation of feasible solutions with the search for the best possible layout design. Suresh and Sahu (1994) used a simulated annealing approach with the single objective of optimization of the "smoothness index," which is concerned with on-time completion of jobs. Shin (1990) created a heuristic that finds a minimum expected total cost by varying cycle times over a feasible range.

The mixed-model, deterministic ALB model is very applicable in just-in-time manufacturing environments, because attempts have been made to find sequences resulting in minimal WIP levels. Kabir and Tabucanon (1995) solved the batch-model assembly line (two or more products) using a multiattribute-based approach. Monden (1993) was concerned with the sequencing of such lines such that stability of parts usage rates would be addressed. Miltenburg (1989) presented a metric used to measure the stability of parts usage rates, as well as a heuristic to minimize the "lumpiness" of raw material usage. Miltenburg and Wijngaard (1994) addressed the U-shape approach, where the flow of assembled units takes on a nonlinear form in order to permit several combinations of activities in work centers, subsequently permitting more design flexibility.

The mixed-model, stochastic ALB problem has received the least amount of attention in recent literature, although it more realistically reflects modern manufacturing environments. As manufacturing flexibility becomes more important, multiple product production lines are more common, so this model relaxes an assumption of a single product type. It is more common for workers in a modern production line to perform many tasks (versus a single task), which is better modeled through stochastic task times rather than deterministic ones. McMullen and Frazier (1997, 1998) approached the problem using several heuristics, as well as simulated annealing techniques, and addressed the complicating feature of task paralleling, which allows multiple workers at a single workstation. With the reality of shorter cycle times and pressures to increase efficiencies, companies often resort to task paralleling to remain competitive. This paper looks at the McMullen and Frazier problem but applies a technique derived from ACO, which is introduced in the following section.

## 2.2. *ACO*

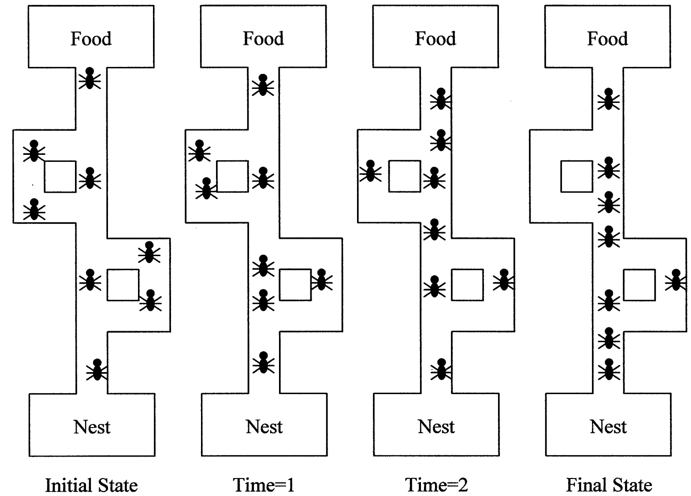
Many aspects of the collective activities of social insects, such as ants, are self-organizing. This means that complex group behavior emerges from the interactions of individuals who exhibit simple behaviors by themselves. Examples of these collective activities among ants are finding food and building nests. The results of self-organization are global in nature, but come about from interactions based entirely on local information. To achieve this, self-organization relies on several components: (i) positive feedback; (ii) negative feedback; (iii) amplification of fluctuations; and (iv) multiple interactions (Bonabeau *et al.*, 1999).

The capabilities of a single ant are very limited compared to those of a colony. In some species, ants are mostly blind and they communicate poorly. But collectively, ants can establish the shortest route between a source of food and their nest, and efficiently move the food to their home. Ants communicate with each other through the use of pheromones. As ants traverse a path, they deposit pheromones. Pheromones are chemical substances that attract other ants, and are deposited by ants on the ground as they travel. Ants move randomly, but when they encounter a pheromone trail, they decide whether or not to follow it. If they do so, they lay down their own pheromone on the trail as well, reinforcing the pathway. The probability that an ant chooses one path over another increases proportional to the amount of pheromone present. The more ants that use a given trail, the more attractive that trail becomes to subsequent ants.

If ants need to decide between a short path and a long path to a source of food, they will first use both paths in equal numbers, laying down pheromone as they travel. But the ants taking the shorter path will return to the nest first. The shorter pathway will then be doubly marked with pheromone, and will be more attractive to those ants that return to the food source. This is illustrated by Fig. 1, which shows a distribution of ants over a set of pathways between a nest and a food source over time. Early on, the ants are equally distributed, but eventually they favor the shorter route.

There is, however, always a chance that an ant will not follow a previous or well-marked trail. This allows for randomness and exploration, which is beneficial in that it allows for the discovery of shorter or alternate pathways, or new sources of food. The pheromones also evaporate over time. Given this, the pheromone will become less detectable on longer trails, since these trails take more time to traverse. The longer trails will hence be less attractive, another benefit to the colony as a whole.

ACO is a meta-heuristic using artificial ants to find desirable solutions to difficult combinatorial optimization problems. The behavior of artificial ants is based on the traits of real ants as described above, plus additional capabilities that make them more effective, such as a memory of past actions. Each ant of the "colony" builds a solution to the problem under consideration, and uses information collected on the problem characteristics and its own performance to change how other ants see the problem. A more detailed explanation of ACO can be found in Dorigo *et al.* (1999). *ACO algorithm* refers to any instantiation of this meta-heuristic. The more informal term *ant algorithm* is used to describe algorithms that, in general, follow ACO guidelines but do not necessarily follow all aspects of the meta-heuristic.




Note: The left-most frame is the initial state. Each subsequent frame is a further point in time, completing with the steady-state in the right-most frame.

**Fig. 1.** Ants presented with alternative pathways (after Goss *et al.* (1989)).

ACO algorithms have already been used with many different problems. Dorigo *et al.* (1996) have addressed the classic traveling salesman problem using pheromone and distances to guide selection of destinations. The closely related vehicle routing problem has been investigated by Bullnheimer *et al.* (1999). An ACO algorithm was developed by Bauer *et al.* (1999) for the single machine total tardiness problem. Colorni *et al.* (1994) looked at the job-shop scheduling problem. Other applications of ACO include space planning (Bland, 1999), graph coloring and partitioning (Costa and Hertz, 1997), and load balancing in telecommunications networks (Schoonderwoerd *et al.*, 1997). An excellent summary of a variety of problems that have been solved using ACO is provided in Dorigo *et al.* (1999). A more general discussion of swarm intelligence techniques and research, which includes ant colony optimization, can be found in Bonabeau *et al.* (1999).

## 3. Methodology

The ALB heuristic presented here works by simulating the travel of artificial ants through each task required by a given line balancing problem. The "ant" selects the task to be added to the current work center via a probabilistic mechanism, the probability of a task being selected by an ant is determined by the level of "pheromone" on the path between the ant and a candidate task. The amount of pheromone on each path is a measure of each path's relative desirability; paths with the largest quantities of pheromone are the most likely to be "chosen" by the ants. After an ant selects a task, whether or not a new work center should be opened is determined via probabilistic means. The above procedure continues until all tasks have been placed into work centers. At this point, the line balancing is complete, and measures of its performance are obtained.

Prior to presenting the details of the ant-based procedure, it is important to define the notation used throughout this section and this is done in Table 1. Also, Figs. 2 and 3 contain a flow chart and pseudo-code, respectively, for the ALB ant-based heuristic that is described in detail in the following pages.

**Table 1.** The notation used in the development of the ALB heuristic

| *Variable* | *Description* |
|---|---|
| $\alpha$ | Work center factor (used to determine whether or not a new work center will be created) |
| $\hat{\sigma}^*_{i,h}$ | Estimated standard deviation of task $i$ for product $h$ |
| $\hat{\sigma}^*_i$ | Estimated standard deviation of task $i$ |
| $\hat{\sigma}_j$ | Estimated standard deviation of work center $j$ |
| $C$ | Cycle time as specified by management |
| $Cost_{\mathrm{L}}$ | Annual labor cost/person |
| $Cost_{\mathrm{M}}$ | Machine cost/year |
| $cv$ | Coefficient of variation as specified by management |
| $d_h$ | Demand (in units) for product $h$ |
| $D_{\mathrm{T}}$ | Total demand for all products |
| $I_i$ | Precedence notation for task $i$ (precedence relations) |
| $L$ | List containing all tasks eligible for assignment into work centers |
| $m_j$ | Machines, or pieces of equipment, required in work center $j$ |
| **M** | An $n \times n$ linkage matrix used to detail precedence relations |
| $metric_i$ | Metric to use with pheromone generation for task $i$ (*objective function related value*) |
| $n_j$ | Number of tasks in work center $j$ |
| $N$ | Total number of tasks for assembly line balancing problem |
| $N_{\mathrm{a}}$ | Number of ants to be simulated |
| $p_j$ | Probability of on-time completion (with respect to the number of available workers) in work center $j$ |
| $ph_i$ | Pheromone associated with task $i$ |
| $R$ | Total number of work centers from solution |
| $S$ | Number of different products to be assembled |
| $tasks_j$ | Number of tasks in work center $j$ |
| $t^*_i$ | Expected duration of task $i$ |
| $t^*_{i,h}$ | Expected duration of task $i$ for product $h$ |
| $t_j$ | Expected duration of all tasks in work center $j$ |
| $u_j$ | Utilization of work center $j$ |
| $\dot{w}_j$ | Integer-adjusted workers required in work center $j$ |
| $w_j$ | Workers required in work center $j$ |

*Note*: it is important to note that $h$ represents a product of interest, $i$ represents a task of interest, while $j$ represents a work center of interest. *Expected* values are intended to serve as means for stochastic analyses.

### *Step 1. Initialization*

The first part of the heuristic is to initialize $L$, $R$, $n_j$, $t_j$, and the **M** matrix to zero. Management also selects values for the new work center factor $\alpha$, as well as one of four strategic approaches. The parameter $\alpha$ provides management with information in order to control the likelihood of establishing a new work center. The parameter $\alpha$ is on the interval [0,1] (values of $\alpha$ approaching unity offer larger probabilities of new work centers being created). Details of the strategic approaches are discussed in Step 3.

Because this methodology addresses a mixed-model scenario for just-in-time production systems, task durations and their associated variances for each task are estimated as follows:

$$t^*_i = \sum_{h=1}^{S} \left(\frac{d_h}{D_{\mathrm{T}}}\right) t^*_{i,h}, \tag{3}$$

$$\hat{\sigma}^*_i = cv \times \sqrt{\sum_{h=1}^{S} \left(\frac{d_h}{D_{\mathrm{T}}}\right)^2 (\hat{\sigma}^*_{i,h})^2}, \tag{4}$$

where

$$D_{\mathrm{T}} = \sum_{h=1}^{S} d_h. \tag{5}$$

These equations are simply estimates of the expected duration and variation of the tasks when stochastic assumptions (and product-mix) are considered. The above three formulae assume the independence of task durations (this is a commonly accepted assumption for assembly line balancing heuristics).

### *Step 2. Determine whether or not a new work center is needed*

If the current work center is empty (it contains no tasks), there is no need to create a new work center. Otherwise, a decision is made whether or not a new work center should be created. The probability of creating a new work center is determined as follows:

$$P(\text{new work center } j) = \alpha / n_j,\ j \in \text{set of existing work centers.} \tag{6}$$

Equation (6) prevents very few large work centers and/or several very small work centers from occurring (the former results in high variable costs, while the latter results in very large fixed costs). A random number on the [0,1] interval is generated, and is used to determine whether or not a new work center is to be opened. In

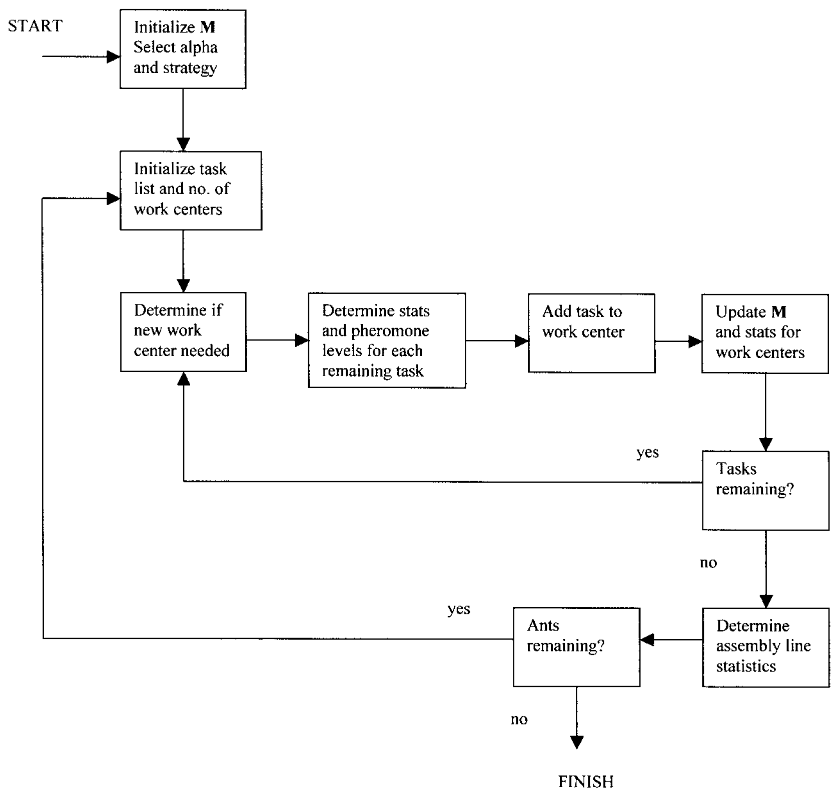



**Fig. 2.** Flow chart for ALB ant heuristic.

this event, all relevant statistics are re-initialized to reflect an "empty" work center. Otherwise, the current work center remains open to accommodate additional tasks. Regardless of the action taken here, the current work center is work center $j$. The value of $\alpha$ is user-specified, and provides the decision-maker with some control over the number of work centers; higher values of $\alpha$ result in a higher probability of new work centers. This is further elaborated upon in the experimentation portion of the research.

```
Begin
    Initialize linkage matrix M
    Select work center factor α
    Select strategic approach
    Do While (iterations < N)
        Initialize L, R, n_j, and t_j
        Do While (tasks remain to be placed in a work center)
            Determine whether or not a new work center is needed
            Construct list of eligible tasks, L, for work center j
            Determine utilization and on-time completion statistics for each task in L
            Determine level of pheromone associated with each task in L using matrix M
            Select task from L to be added to work center j
            Update statistics for work center j
            Update matrix M
        End {do while}
        Compute assembly line design performance statistics
        Track "best" assembly line design
        iterations = iterations + 1
    End {do while}
End.
```

**Fig. 3.** Pseudo-code for ALB ant heuristic.

***Step 3. Selection of tasks***

Once initialization is complete, tasks can be selected for addition to each work center (an aggregation of separable tasks). Prior to task selection, a list of eligible tasks, $L$, is constructed. Task $i$ will be added to $L$ if and only if the following conditions exist:

- all of task $i$'s predecessors have already been assigned to work centers;
- task $i$ has not already been assigned to a work center.

For each task having membership in $L$, temporary values of $u_j$ and $p_j$ are determined. The utilization of work center $j$ ($u_j$) if task $i$ were to be added is given as:

$$u_j = \frac{w_j}{\dot{w}_j}, \tag{7}$$

where

$$w_j = \frac{1}{C}(t_j + t_i^*), \tag{8}$$

and

$$\dot{w}_j = 1 + \text{int}(w_j). \tag{9}$$

The probability of all tasks being completed on-time ($p_j$), with respect to the number of available workers) if task

$i$ were to be added to work center $j$ is estimated as follows:

$$p_j = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{Y} \exp(-0.5z^2)\,\mathrm{d}z, \tag{10}$$

where

$$Y = \frac{(\dot{w}_j - w_j) \times C}{\hat{\sigma}_j}, \tag{11}$$

and

$$\hat{\sigma}_j = \sqrt{\hat{\sigma}_j^2 + \hat{\sigma}_i^{*2}}. \tag{12}$$

The assumption here is that if there is enough available labor in the work center in comparison to expected task durations, the job should be completed on-time. Otherwise, the job will not be completed on-time. It is also assumed that the labor requirement for each work center follows a normal distribution, as indicated by Equation (10).

After utilization and on-time completion statistics have been calculated for each task in $L$, a metric is calculated and used to construct the pheromone trail. The metric is determined by the strategic approach chosen by management. These strategic approaches are based on user-defined objectives with the intent of optimizing some sort of performance-related objective, such as efficiency, on-time completion or design cost (sum of labor and equipment costs). Four metrics for the strategic approaches are shown in Table 2.

The level of pheromone associated with task $i$ is determined as follows:

$$ph_i = \left( metric_i \Big/ \sum_i^n metric_i \right) + \left( \mathbf{M}(i, I_i) \Big/ \sum_i^n \mathbf{M}(i, I_i) \right). \tag{13}$$

The first component of the equation above is straightforward, the relative desirability of task $i$'s *metric* value, as compared to other tasks in $L$. The second component requires more explanation. The **M** matrix is a historical representation (based on the number of ants simulated) of how often task $i$ has been selected to enter the current work center after a certain immediate predecessor, and $I_i$ is a precedence notation. **M** counts how many times ants have chosen to travel from one task to another, and is ultimately used to make pheromone determinations. For example, given two tasks, task 1 and task 2, if task 2 is selected to enter the current work center immediately after task 1 (based upon the value of $ph_2$ above), then the value for **M**(2,1) is incremented by one. This matrix component of the pheromone calculation is intended to exploit the relationship between tasks being selected after certain predecessors have been placed into the work centers. If, again given task 1 and task 2, task 1 is a frequent immediate predecessor of task 2 in the work center of interest, then this historical component adds a large quantity of pheromone to the equation above. The second component of this equation is intended to give historical input to the task selection choice. The probability of this historical relationship, or linkage (and other competing relationships, or linkages) is then adjusted by incrementing the value in **M** for the selected task to follow the preceding task. It is also important to note that if the current work center is empty (there are no tasks present), then the second part of the equation is zero, for the simple reason that there are no immediate predecessors from which to establish linkage.

**Table 2.** Task selection strategies

| *Strategic approach* | Metric$_i$ |
|---|---|
| ANT-1 | $u_i$ |
| ANT-2 | $p_i$ |
| ANT-3 | $u_i \times p_i$ |
| ANT-4 | $u_i \times (1 - p_i)$ |

This "dual-component" pheromone calculation results in a heuristic that utilizes concepts from both previous ALB solution methods and newer ant techniques such as ACO. The first component assures that traditional ALB performance measures are taken into account (such as efficiency, on-time completion or design cost). The second component utilizes the historical aspects of task selection, that is, preference is given to those activities having been most frequently selected in the past (this is intended to "reinforce" popular historical choices, as is done with updating in ACO approaches). The second component of Equation (13) is intended to emulate the pheromone trail updating via an ACO approach. The dynamics of the problem at hand make it difficult to perform local and global updating that are a part of the traditional ACO problem, which is typically described in the context of the traveling salesman problem. Subsequently, the second component of Equation (13) is used to emulate pheromone reinforcement, making popular historical choices more likely to occur. It is emphasized here that an ant selects tasks for work centers during its "journey" through the problem; its journey is complete when all tasks have been assigned to work centers.

The pheromone value for task $i$ presented above ($ph_i$) is the probability of task $i$ being selected for the current work center of interest (work center $j$). Monte Carlo simulation is then used to select the task from $L$ to be added to the current work center $j$. The Monte Carlo simulation works by providing each task $i$ with a probability of being selected equal to $ph_i$ divided by the sum of all $ph_i$ calculations (all $i \in L$). These probabilities are then converted into cumulative probability distributions, which are then converted into random number ranges. From there, random numbers are generated in the [0,1] interval to select the tasks. Therefore, tasks with higher pheromone levels have a higher probability of being selected, but the next task to be selected depends in part on chance.

***Step 4. Update relevant statistics***

After the selection of the task from $L$ to be added to work center $j$, the following statistics are permanently updated to reflect the addition of task $i$: $t_j$, $\hat{\sigma}_j$, $u_j$ and $p_j$. Specifically:

$$t_j = \sum_{i=1}^{n_j} t_i^*,\ i \in \text{set of all tasks allocated to work center } j, \tag{14}$$

$$\hat{\sigma}_j = \frac{1}{C}\sqrt{\sum_{i=1}^{n_j} \hat{\sigma}_i^{*2}},\quad i \in \text{set of all tasks allocated to work center } j. \tag{15}$$

Values for $u_j$ and $p_j$ are calculated the same way as they are above, as shown in Equations (7) and (10) respectively. The only difference here is that the values of $t_j$ and $\hat{\sigma}_j$ reflect the results of Equations (14) and (15) above. The linkage matrix **M** is also revised to reflect the addition of task $i$ to work center $j$ (assuming work center $j$ is non-empty). Specifically, the value $\mathbf{M}(i, I_i)$ is incremented by one.

***Step 5. Continuation until completion***

The simulated ant continues the steps detailed in Steps 2, 3 and 4 until all tasks have been placed in work centers. When the ant has completed its visit of all $n$ tasks, the process of placing tasks in the work centers is complete. The result is an assembly line layout.

***Step 6. Compute assembly line statistics***

Upon completion of the simulated ant behavior, the following statistics are collected pertaining to the quality of the assembly line design:

$U$ = utilization of assembly line layout;
$P$ = probability of all work centers being completed on-time;
$Comp$ = composite measure of utilization and probability of on-time completion;
$Cost$ = design cost associated with layout.

The utilization ($U$) of an entire layout is the ratio of the required minimum number of workers to the actually assigned number of workers, and can be formally written as:

$$U = \left(\frac{1}{C}\right)\sum_{i=1}^{n} t_i^* \Big/ \sum_{j=1}^{R} \dot{w}_j. \tag{16}$$

The probability of on-time completion for all work centers in the layout is estimated as follows:

$$P = \prod_{j=1}^{R} p_j. \tag{17}$$

The composite measure of both system utilization and on-time completion is:

$$Comp = U \times P \tag{18}$$

The design cost of the layout, which considers both the number of personnel required as well as the number of machines required (a linear combination of both labor and equipment costs), is the following:

$$Cost = Cost_{\mathrm{L}} \times \sum_{j=1}^{R} \dot{w}_j + Cost_{\mathrm{M}} \times \sum_{j=1}^{R} (tasks_j \times \dot{w}_j). \tag{19}$$

The major assumptions for this design cost measurement are that labor costs for each employee are \$30 000 per year ($Cost_{\mathrm{L}}$), and that each machine has an annual cost of \$3000 ($Cost_{\mathrm{M}}$) (these values seem to reflect reasonable assumptions of the annual labor and annual equipment costs).

***Step 7. Objective function***

As the six steps above are executed, an assembly line layout is obtained, and its subsequent performance statistics are captured. Each time the performance statistics are captured, they are compared with the most desired performance statistics found thus far. For this research, there are four objective functions employed, (See Table 3) and they are analogous to the four strategies described in Step 3 of the Methodology section.

If the objective function value for the layout of the current ant is found superior to the "best one" in memory, the current layout and its relevant statistics become the "best." This procedure repeats itself for $N_{\mathrm{a}}$ iterations, and the "best" solution found, in terms of objective function value, is reported.

## 4. Experimentation

To assess the desirability of the ant-based approach presented above, experimentation was performed to compare the ant approach with several others, including work-center loading approaches and simulated annealing approaches. Specifically, the solutions obtained from all of these heuristic approaches are used as layouts for simulated production runs of a just-in-time production assembly line.

**Table 3.** Objective function strategies

| *Objective function label* | *Objective* |
|---|---|
| ANT-1 | max($U$) |
| ANT-2 | max($P$) |
| ANT-3 | max($Comp$) |
| ANT-4 | min($Cost$) |

**Table 4.** Description of the heuristics used in the comparison

| *Heuristic label* | *Heuristic description* |
|---|---|
| T-1 | Select task with maximum incremental utilization |
| T-2 | COMSOAL (random selection of tasks) |
| T-3 | Select task with random duration |
| T-4 | Select task with shortest duration |
| T-5 | Select task with minimum incremental utilization |
| T-6 | Select task providing minimum lateness probability |
| T-7 | Composite of T-5 and T-6 |
| LEX-1 | Lexicographic task selection |
| WC-1 | Single, mega-work center |
| WC-2 | Single work center for each task |
| PREC-1 | Select task having fewest followers |
| PREC-2 | Select task having fewest immediate followers |
| PREC-3 | Select first available task |
| PREC-4 | Select last available task |
| PREC-5 | Select task having most followers |
| PREC-6 | Select task having fewest followers |
| RPWT | Select task having highest ranked positional weight |
| SA-1 | Simulated annealing: minimize design cost |
| SA-2 | Simulated annealing: minimize smoothness index |
| SA-3 | Simulated annealing: minimize system lateness |
| SA-4 | Simulated annealing: composite of SA-1 and SA-2 |
| SA-5 | Simulated annealing: composite of SA-1 and SA-2 |
| SA-6 | Simulated annealing: composite of SA-1 and SA-2 |
| ANT-1 | ACO: maximize utilization |
| ANT-2 | ACO: maximize on-time completion probability |
| ANT-3 | ACO: composite of ACO-1 and ACO-2 |
| ANT-4 | ACO: minimize design cost |

*Note*: The differences between SA-4, SA-5 and SA-6 are that they have varying weighting between the objectives of SA-1 and SA-2. TASK (task related), PREC (precedence related), RPWT (ranked positional weight technique), SA (simulated annealing related).

Information regarding the efficiency, on-time completion, and design cost of each layout obtained via the ant approach is compared with the same outputs for the other heuristic approaches. Details of these other approaches are contained in Table 4. The interested reader can learn details pertaining to the comparison heuristics by referring to Arcus (1966) for T-2, Helgeson and Birnie (1961) for RPWT, McMullen (1995) and McMullen and Frazier (1997) for the remaining T-x and PREC-x heuristics. Readers can also refer to McMullen and Frazier (1998) for the WC-x and SA-x heuristics. LEX-1 is explained in Gaither and Frazier (1999). In essence, these approaches are previously published heuristics that have been used to address the assembly line problem with the same set of assumptions and complexities of interest here.

Six problems were addressed in this research, ranging in size from 21 to 74 tasks, as shown in Table 5. These varied problems from the literature are intended to permit the authors to make general claims about the performance of the presented methodology.

**Table 5.** Heuristic parameters

| *Problem size* | *ANT-1 $\alpha$* | *ANT-2 $\alpha$* | *ANT-3 $\alpha$* | *ANT-4 $\alpha$* | $N_a$ |
|---|---|---|---|---|---|
| 21 Tasks | 0.5 | 0.4 | 0.1 | 0.5 | 250 |
| 25 Tasks | 0.5 | 0.9 | 0.2 | 0.5 | 250 |
| 29 Tasks | 0.5 | 0.6 | 0.3 | 0.5 | 250 |
| 40 Tasks | 0.5 | 0.5 | 0.2 | 0.7 | 500 |
| 45 Tasks | 0.5 | 0.6 | 0.2 | 0.6 | 500 |
| 74 Tasks | 0.5 | 0.3 | 0.3 | 0.6 | 1000 |

### 4.1. *Heuristic parameters*

As mentioned in the methodology section, the new work-center factor, $\alpha$, is a parameter that is controlled by the decision-maker. Experimentation was used to find values of $\alpha$ which deliver the most promising objective function values (for each problem and approach, $\alpha$ was varied between zero and one in increments of 0.025, and 500 runs were used at each value of $\alpha$). Table 5 shows the selected values of $\alpha$ for each of the four strategic approaches for each of the six respective problems. Also presented in the table is the number of ants ($N_a$) that were simulated for the heuristics.

### 4.2. *Simulation parameters and assumptions*

As stated earlier, solutions obtained via the search heuristics are used as layouts for simulated production runs. The SLAM simulation language and user-written FORTRAN inserts are used to perform these production simulations. It was also assumed that multiple products are required for various problems; this is referred to as a mixed-model ALB problem. For these problems, the job sequencing approach used is from Miltenburg (1989), which optimizes the stability of material usage rates. Table 6 shows parameters used for each of the six different problems along with mixed-model information.

A brief explanation is needed to clarify Table 6. For each problem, the number of unique products is listed, followed by product-mix details and the number of "build-up" minutes for the production simulation. For example, the 40-task problem has three different items demanded. Three units of product 1 are demanded, two units of product 2 are demanded, and a single unit of product 3 is demanded (3,2,1). Also, when the simulated production run is executed, all statistics are "cleared out" and reset after 3000 simulated minutes. This resetting of simulation statistics is done so that any biases associated with system build-up are avoided.

After the resetting of simulation statistics, the production run is simulated for an additional 3000 minutes for all problems. At the completion of each simulation run, statistics of interest are collected, and the process of resetting statistics and running an additional 3000 minutes is repeated 24 more times. Twenty-five simulated production runs are done to

**Table 6.** Simulation details

| *Problem size* | *Different products* | *Product-mix* | *Simulation build-up time (minutes)* | *Literature source* |
|---|---|---|---|---|
| 21 Tasks | 1 | 1 | 500 | Tonge (1965) |
| 25 Tasks | 1 | 1 | 1000 | McMullen and Frazier (1997) |
| 29 Tasks | 2 | 2,1 | 3000 | Buxey (1974) |
| 40 Tasks | 3 | 3,2,1 | 3000 | McMullen and Frazier (1997) |
| 45 Tasks | 2 | 2,1 | 3000 | Kilbridge and Wester (1962) |
| 74 Tasks | 4 | 4,2,1,1 | 3000 | McMullen and Frazier (1997) |

ensure that reliable estimates of heuristic performance measures are obtained. These replications are important (as are the number of *ants* chosen) because it accounts for the fact that the decisions made during the search can cause differing results from replication to replication (or from *ant* to *ant*). Multiple replications (and multiple *ants*) are then required to assure a performance estimate representative of all outcomes.

Another assumption is that task durations are uniformly distributed between 2 and 10 minutes with a 75% probability, and that task durations are uniformly distributed between 10 and 15 minutes with a 25% probability. Another assumption is that a pull system is utilized for the simulated production runs; a unit being assembled is not moved through the system until it is requested by the downstream worker. This pull system is a distinguishing characteristic of just-in-time production systems.

### 4.3. *Output performance measures and research questions*

Several measures of the assembly line layout are of interest here: average flow-time, average work-in-process inventory level, probability of jobs being completed on-time, system efficiency, and design cost. Average Flow-Time (FT) is simply the average amount of time a job spends in the system. Average work-in-process inventory level (WIP) is the average number of units in the production system at any given time. Small values of both FT and WIP are desired, and are important measures of the desirability of just-in-time systems. The probability of jobs being completed on time (POT) is the percentage of time that jobs make it through all work centers within their desired cycle time; this is a measure of schedule compliance and high values of POT are desired. There are several ways to measure system efficiency, but the chosen metric here is the ratio of the desired cycle time obtained to the actual cycle time. This measure of efficiency is referred to as the Cycle Time Ratio (CTR), and higher levels are desired. These described performance measures are obtained via the production simulation runs as described in Section 4.2. Design cost (COST) is simply the cost of both human resources and machines; this computation is detailed in Equation (19). It is desired to minimize COST.

The primary research question of interest here is quite straightforward: how well do the ant-based approaches presented here compare against the other heuristics? To answer this general question, univariate ANOVA is used along with comparisons of the mean performance measures defined above. There is a separate ANOVA and means comparison for each of the output performance measures across the six different problems. It is also desired to explore the effect (if any) of the problem size on the results.

## 5. Experimental results and discussion

Prior to presenting actual results, it is important to note that there is a direct correlation between average FT and WIP ($\rho = 1.000$). These two variables are also inversely related to the cycle time ratio defined above ($\rho = -0.939$). In essence, then, high values of CTR result in low values of FT and WIP, and *vice versa*. Subsequently, WIP and FT are removed from the analysis, and their performance is explained by CTR. This results in a more parsimonious interpretation. As a result of the omission of FT and WIP, there are three output performance measures for examination here: (i) the probability of all jobs being completed on-time; (ii) the cycle time ratio; and (iii) the composite cost of both personnel and equipment (i.e., POT, CTR, and COST). Table 7 shows means and standard deviations of POT, CTR and COST for the four ANT strategies presented here (as obtained from the simulation described above), along with the other 23 comparison heuristics from the literature. Along with these means and standard deviations, a univariate $F$ statistic and associated $p$-values are presented, the result of ANOVA to determine the effect of the heuristic on the performance measure of interest. In other words, ANOVA is used to see if differences in means of the performance measures of interest are the result of the different search heuristics.

From inspection of Table 7, it is clear that the choice of heuristic has an effect on the performance measures of interest; ANOVA shows all $p$-values are $<0.001$. It is then desired to determine the quality of the ant-based heuristic approaches relative to the other approaches. Examination of means assists with this, but 95% confidence interval plots (Figs. 4, 5, and 6) are employed for a graphical explanation.

**Table 7.** Means (standard deviations) and rankings of POT, CTR and COST for the investigated heuristics

| *Heuristic* | *POT mean (Std)* | *CTR mean (Std)* | *Cost mean (Std)* | *POT rank* | *CTR rank* | *Cost rank* |
|---|---|---|---|---|---|---|
| T-1 | 0.7306 (0.0347) | 0.8331 (0.0844) | 1225 500 (594 614) | 23 | 23 | 2 |
| T-2 | 0.6650 (0.0474) | 0.8571 (0.0667) | 1283 500 (690 484) | 27 | 20 | 3 |
| T-3 | 0.7549 (0.0398) | 0.8133 (0.1124) | 1326 000 (775 758) | 16 | 25 | 10 |
| T-4 | 0.7061 (0.0840) | 0.8712 (0.0598) | 1337 000 (739 376) | 25 | 14 | 12 |
| T-5 | 0.7423 (0.0675) | 0.8364 (0.1570) | 1491 000 (821 761) | 20 | 22 | 17 |
| T-6 | 0.7465 (0.0544) | 0.8865 (0.0626) | 1475 000 (895 073) | 18 | 12 | 15 |
| T-7 | 0.7593 (0.0582) | 0.8961 (0.0548) | 1479 000 (930 356) | 15 | 11 | 16 |
| LEX-1 | 0.7462 (0.0504) | 0.8736 (0.0676) | 1316 000 (666 050) | 19 | 13 | 8 |
| WC-1 | 0.7940 (0.1223) | 0.9238 (0.1873) | 4053 000 (2482 792) | 9 | 8 | 25 |
| WC-2 | 0.9076 (0.0390) | 0.9065 (0.0908) | 1611 500 (817 819) | 2 | 9 | 21 |
| PREC-1 | 0.7531 (0.0483) | 0.8584 (0.0829) | 1300 500 (647 958) | 17 | 19 | 5 |
| PREC-2 | 0.7333 (0.0481) | 0.8609 (0.0754) | 1319 500 (721 861) | 22 | 18 | 9 |
| PREC-3 | 0.7758 (0.0310) | 0.8679 (0.0583) | 1328 000 (730 017) | 14 | 15 | 11 |
| PREC-4 | 0.7765 (0.0765) | 0.8122 (0.1448) | 1304 500 (707 897) | 13 | 26 | 6 |
| PREC-5 | 0.7851 (0.0893) | 0.8670 (0.0969) | 1354 000 (664 928) | 12 | 16 | 14 |
| PREC-6 | 0.7903 (0.0696) | 0.8485 (0.0866) | 1286 000 (651 108) | 11 | 21 | 4 |
| RPWT | 0.7939 (0.0962) | 0.8629 (0.1098) | 1313 500 (690 246) | 10 | 17 | 7 |
| SA-1 | 0.6970 (0.0455) | 0.7780 (0.1338) | 1204 000 (600 305) | 26 | 27 | 1 |
| SA-2 | 0.7249 (0.0824) | 0.9030 (0.0544) | 1868 500 (1246 068) | 24 | 10 | 23 |
| SA-3 | 0.9343 (0.0611) | 0.9899 (0.0168) | 1672 000 (843 045) | 1 | 2 | 22 |
| SA-4 | 0.9038 (0.0542) | 0.9742 (0.0330) | 1545 000 (749 471) | 4 | 6 | 19 |
| SA-5 | 0.9066 (0.0470) | 0.9753 (0.0381) | 1557 500 (795 975) | 3 | 5 | 20 |
| SA-6 | 0.8940 (0.0713) | 0.9667 (0.0427) | 1513 000 (769 259) | 5 | 7 | 18 |
| ANT-1 | 0.7372 (0.1608) | 0.9883 (0.0229) | 4883 000 (4895 447) | 21 | 3 | 27 |
| ANT-2 | 0.8738 (0.1205) | 0.9936 (0.0131) | 2785 500 (2075 500) | 6 | 1 | 24 |
| ANT-3 | 0.8656 (0.1125) | 0.9868 (0.0160) | 4489 000 (4510 321) | 7 | 4 | 26 |
| ANT-4 | 0.8059 (0.0695) | 0.8267 (0.1647) | 1345 074 (680 207) | 8 | 24 | 13 |
| Univ. *F* | 5.59 ($p < 0.001$) | 2.84 ($p < 0.001$) | 2.44 ($p < 0.001$) | | | |

The on-time performance resulting from two of the ant-based approaches is competitive with the best of the other heuristics approaches used for comparison purposes. Specifically, ANT-2 and ANT-3 provide relatively desirable results in terms of on-time performance (ANT-4 is also somewhat competitive here). This makes sense due to the fact that the objective of ANT-2 is to maximize the system probability of on-time completion ($P$) via work-center on-time completion ($p_j$). ANT-3 attempts the same thing but in a composite form along with system utilization ($U \times P$) via the work center composite form ($u_j \times p_j$). The comparison heuristics performing desirable with regard to POT are WC-2 and SA-3, SA-4, SA-5 and SA-6. WC-2 has a single task performed in each work center, which provides the ability for less clutter, and subsequently, less lateness. Unfortunately, WC-2 has the side effect of a higher design cost (as compared to several others) due to the labor costs associated with a single task in each work center. However, ANT-1 performs poorly with regard to on-time completion. ANT-1 seems to provide high system utilization (it attempts to minimize $u_j$ and subsequently $U$) at the associated expense of high system lateness. ANT-1 suggests that

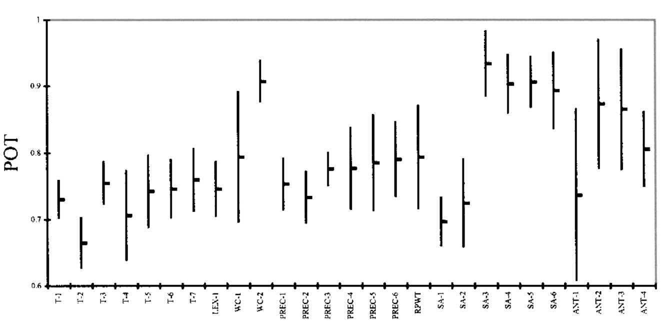



**Fig. 4.** The 95% confidence intervals for POT.

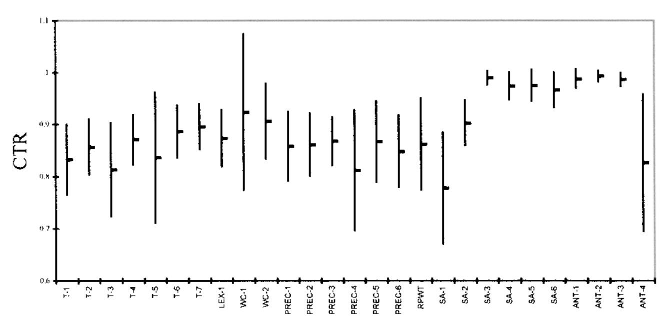



**Fig. 5.** The 95% confidence intervals for CTR.

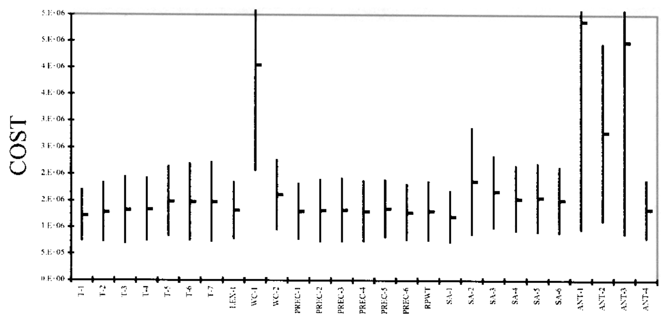



**Fig. 6.** The 95% confidence intervals for COST.

there appears to be a trade-off between on-time completion and system utilization.

In terms of system efficiency, which is measured by the CTR, three of the four ant-based approaches perform well: ANT-1, ANT-2, and ANT-3. It is understandable why ANT-1 and ANT-3 perform well, optimization of system utilization is sought via both of these objective functions. The strong performance of ANT-2 in terms of system efficiency seems to be a welcome side effect of a strategy concerned with on-time completion. Four of the six simulated annealing approaches (SA-3–SA-6) also perform well with regard to system efficiency. ANT-4 performs relatively poorly with regard to CTR, its objective intended to minimize design cost and did not result in efficient layouts. The reason for this is believed to lie in the fact that the relatively small number of machines offered by ANT-4 results in some bottlenecking, which clutters production.

Despite the fact that ANT-4 does not perform well regarding on-time completion or system efficiency, it does perform competitively regarding design cost. This should not be surprising given that the essential strategy behind ANT-4 is to in fact minimize the composite cost involving both personnel and equipment. ANT-4 is not alone in its strong performance regarding desirable level of design cost, some task oriented approaches (T-1–T-4) perform well, as well as LEX-1 and some precedence related approaches (PREC-1–PREC-4 and PREC-6). The RPWT and SA-1 approaches also result in relatively desirable levels of design cost. It is also important to note that the other ant-based approaches (ANT-1, ANT-2 and ANT-3) are poor performers in terms of design cost. Their intent of delivering desirable performances in terms of utilization and on-time performance does not simultaneously result in manageable levels of human and equipment resources.

Another finding worth noting is that a generally poor performance with respect to these ANT approaches shows a large variation for the objective function measure, while a generally strong performance shows little variation. Figs. 4, 5 and 6 graphically support this claim.

The analysis also shows that the three performance measures of interest here are sensitive to the problem set. Table 8 details this finding.

In general, Table 8 shows that the larger problem sets result in higher costs, more lateness, and lost efficiency. In other words, as the problems get larger, there are more opportunities for clutter and inefficiency. These results are not surprising; it is emphasized again here that several different problems are used for analysis in this research so that general findings can be claimed.

## 6. Concluding comments

The ANT-1 approach, which is intended for maximum system utilization, delivers strong performance in the form of efficiency, measured here in terms of cycle time performance. Unfortunately, it does not perform well regarding the other two performance measures. Because it does offer efficiency at the expense of other measures, it could be considered when labor costs are low. It should also be considered when inventory clutter costs are high, recall that cycle time performance is also used here to "speak for" WIP levels and flow-time. The ANT-2 and ANT-3 strategies are strong performers with respect to on-time completion and efficiency, but perform poorly in terms of design cost; these two strategies should be used when labor and equipment costs are tolerable, and/or when desirable performance pertaining to the less quantifiable performance measures is imperative. The decision to use ANT-4 depends on the costs of personnel and equipment. When these costs are high, ANT-4 is a desirable strategy. When on-time performance, system efficiency, tolerable levels of WIP and flow-time are required, the decision-maker is better served using one of the other ant-based strategies.

**Table 8.** Means (and standard deviations) of the performance measures by the problem sets

| *Problem set* | *POTC* | *CTR* | *COST* |
|---|---|---|---|
| 21 Tasks | 0.9393 (0.0393) | 0.9978 (0.0048) | 1135 500 (356 310) |
| 25 Tasks | 0.7473 (0.1704) | 0.9738 (0.0434) | 1423 500 (379 871) |
| 29 Tasks | 0.7540 (0.1196) | 0.9754 (0.0334) | 1917 000 (599 651) |
| 40 Tasks | 0.8549 (0.0522) | 0.8676 (0.1937) | 2721 000 (1176 939) |
| 45 Tasks | 0.8855 (0.0553) | 0.9694 (0.0442) | 3897 758 (1470 338) |
| 74 Tasks | 0.7582 (0.0833) | 0.9353 (0.0749) | 9222 750 (4836 794) |
| *F*-statistic | 68.91 | 26.95 | 199.34 |

Overall, the ant-system related heuristic studied in this research shows mixed results in terms of solving the ALB problem with parallel workstations, stochastic task durations, and mixed-models. If efficient cycle time performance is the goal, then three of the four ant-related heuristics (all but ANT-4) outperform the other 23 comparison heuristics tested. If lower cost is the objective, then only one ant-related heuristic (ANT-4) performs as well as the comparison heuristics. In terms of on-time completion, two ant-heuristics (ANT-2 and ANT-3) perform well, but not as well as several of the heuristics based on simulated annealing. The ant methodologies also seem to perform better for smaller problem sizes.

This paper has shown how ant-system related techniques can be applied to the solution of the ALB problem. A logical step in the refinement of the heuristic would be to investigate using a multiplicative operator instead of an additive one when determining pheromone levels. The heuristic might also implement pheromone evaporation, and take into account the quality of each solution when determining pheromone levels.

Researchers are just beginning to apply what is known about collective ant behaviors to heuristic design. The studies that have been done to date, along with the research in this paper, show the potential of this approach to effectively find good solutions to many types of practical problems that would otherwise remain difficult to solve. An opportunity for future research specific to the topic at hand would be to use this methodology to address the U-shaped ALB problem with assumptions similar to the ones addressed here. Looking at a more general problem, the simulation of social insect behavior could also be used to determine crew assignments, in addition to many other applications.

## Acknowledgement

The authors wish to thank the anonymous referees and the editor for their constructive feedback, which resulted in the subsequent improvement of this paper.

## References


Anderson, E.J. and Ferris, M.C. (1994) Genetic algorithms for combinatorial optimization: the assembly line balancing problem. *ORSA Journal on Computing*, **6**, 161–173.

Arcus, A.L. (1966) COMSOAL: a computer method of sequencing operations for assembly lines. *International Journal of Production Research*, **4**, 259–277.

Bauer, A., Bullnheimer, B., Hartl, R.F. and Strauss, C. (1999) An ant colony optimization approach for the single machine total tardiness problem, *Proceedings of the 1999 Congress on Evolutionary Computation*, pp. 1445–1450.

Bland, J.A. (1999) Space-planning by ant colony optimisation. *International Journal of Computer Applications in Technology*, **12**, 320–328.

Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999) *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, NY.

Bullnheimer, B., Hartl, R.F. and Strauss, C. (1997) *Applying the Ant System to the Vehicle Routing Problem*, presented of the 2nd Meta Heuristic International Conference, Sophia-Antipolis, France.

Buxey, G.M. (1974) Assembly line balancing with multiple stations. *Management Science*, **20**, 1010–1021.

Colorni, A., Dorigo, M., Maniezzo, V. and Trubian, M. (1994) Ant system for job-shop scheduling. *Belgian Journal of Operations Research*, **34**, 39–53.

Costa, D. and Hertz, A. (1997) Ants can colour graphs. *Journal of the Operational Research Society*, **48**, 295–305.

Dorigo, M., Di Caro, G. and Gambardella, L. M. (1999) Ant algorithms for discrete optimization. *Artificial Life*, **5**, 137–172.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996) The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, **26**, 29–41.

Easton, F., Faaland, B., Klastorin, T.D. and Schmitt, T. (1989) Improved network based algorithms for the assembly line balancing problem. *International Journal of Production Research*. **27**, 1901–1915.

Gaither, N. and Frazier, G.V. (1999) *Production and Operations Management*, 8th edn, Southwestern Publishing Company, Cincinnati, OH.

Ghosh, S. and Gagnon, R. (1989) A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *International Journal of Production Research*, **27**, 637–670.

Goss, S., Aron, S., Deneubourg, J.L. and Pasteels, J.M. (1989) Self-organized short-cuts in the Argentine ant. *Naturwissenschaften*, **76**, 579–581.

Gutjahr, A.L. and Nemhauser, G.L. (1964) An algorithm for the line balancing problem. *Management Science*, **11**, 308–315.

Helgeson, W.B. and Birnie, D.P. (1961) Assembly line balancing using the ranked positional weight technique. *The Journal of Industrial Engineering*, **12**, 394–398.

Hoffman, T.R. (1992) Eureka: a hybrid system for assembly line balancing. *Management Science*, **38**, 39–47.

Kabir, A. and Tabucanon, M.T. (1995) Batch-model assembly line balancing: a multiattribute decision making approach. *International Journal of Production Economics*, **41**, 193–201.

Kilbridge, M.D. and Wester, L. (1962) The balance delay problem. *Management Science*, **8**, 69–84.

Leu, Y-Y., Matheson, L.A. and Rees, L.P. (1994) Assembly line balancing using genetic algorithms with heuristic-generated initial populations and multiple evaluation criteria. *Decision Sciences*, **25**, 581–605.

Lyu, J. (1997) A single-run optimization algorithm for stochastic assembly line balancing problems. *Journal of Manufacturing Systems*, **16**, 204–210.

Malakooti, B. (1991) A multiple criteria decision making approach for the assembly line balancing problem. *International Journal of Production Research*, **29**, 1979–2001.

Malakooti, B. (1994) Assembly line balancing with buffers by multiple criteria optimization. *International Journal of Production Research*, **32**, 2159–2178.

McMullen, P.R. (1995) A simulation approach to solving the type *i* assembly line balancing problem for mixed-models with stochastic task durations. Doctoral Dissertation: University of Oregon, Eugene, OR.

McMullen, P.R. and Frazier, G.V. (1997) A heuristic for solving mixed-model line balancing problems with stochastic task durations and parallel stations. *International Journal of Production Economics*, **51**, 177–190.

McMullen, P.R. and Frazier, G.V. (1998). Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstations. *International Journal of Production Research*, **36**, 2717–2741.

Miltenburg, G.J. (1989) Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science*, **35**, 192–207.

Miltenburg, G.J. and Wijngaard, J. (1994) The U-line line balancing problem. *Management Science*, **40**, 1378–1388.

Monden, Y. (1993) *Toyota Production System*, 2nd edn, Industrial Engineering and Management Press, Institute of Industrial Engineers, Norcross, GA.

Nkasu, M.M. and Leung, K.H. (1995) A stochastic approach to assembly line balancing. *International Journal of Production Research*, **33**, 975–991.

Sarin, S.C., Erel, E. and Dar-El, E.M. (1999) A methodology for solving single-model, stochastic assembly line balancing problem. *Omega*, **27**, 525–535.

Schoonderwoerd, R., Holland, O., Bruten, J. and Rothkrantz, L. (1997) Ant-based load balancing in telecommunications networks. *Adaptive Behavior*, **5**, 169–207.

Shin, D. (1990) An efficient heuristic for solving stochastic assembly line balancing problems. *Computers & Industrial Engineering*, **18**, 285–195.

Sprecher, A. (1999) A competitive branch-and-bound algorithm for the simple assembly line balancing problem. *International Journal of Production Research*, **37**, 1787–1816.

Suresh, G. and Sahu, S. (1994) Stochastic assembly line balancing using simulated annealing, *International Journal of Production Research*, **32**, 1801–1810.

Tarasewich, P. and McMullen, P.R. (2002) Swarm intelligence: power in numbers. *Communications of the ACM*, **45**, 62–67.

Tonge, F.M. (1965) Assembly line balancing using probabilistic combinations of heuristics. *Management Science*, **11**, 727–735.

## Biographies

Patrick R. McMullen is an Associate Professor of Management at Wake Forest University's Babcock Graduate School of Management. His Ph.D. is from the University of Oregon. He has held teaching positions at the University of Oregon, the University of Maine, Harvard University, and Auburn University. His research interests lie in the areas of applied optimization and search heuristics.

Peter Tarasewich is an Assistant Professor at Northeastern University's College of Computer Science. His Ph.D. is from the University of Connecticut. He has held prior teaching positions at the University of Connecticut and the University of Maine. His research interests lie in the areas of telecommunications, e-commerce, applied optimization and search heuristics.

*Contributed by the Engineering Design Department.*