

Do ants paint trucks better than chickens? Markets versus response thresholds for distributed dynamic scheduling

Oran Kittithreerapronchai

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205
gte094z@prism.gatech.edu

Carl Anderson

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205
carl@isye.gatech.edu (corresponding author)

Abstract- We studied the dynamic allocation of trucks to paint booths, contrasting two previously proposed schemes in which booths bid against each other for trucks: one based on markets and the other ant-inspired response thresholds. We explore parameter space for several system performance metrics and find that this system is surprisingly easy to optimize and that a number of parameters can be eliminated. We investigate two different threshold reinforcement schemes that give rise to booth specialization and also examine variations of the breaking tie rules that decide among booths when two or more place identical, highest bids for a particular truck. We find that the threshold reinforcement scheme usually used in response threshold applications (local update) fares worse than one with global update of thresholds, and that breaking tie rules previously proposed can be simplified without loss of system performance.

1 Introduction

As hard as we try to control them, manufacturing processes rarely operate in the regular, predictable and clockwork manner that we imagine or desire (e.g. Smith 1776 and Taylor 1911). Not only must the process contend with internal perturbations, e.g. worker, software and machine failure, but at some point, manufacturing processes have to interact with the external world, a world full of perturbations and unpredictability. Scheduling, therefore, can be a huge challenge to say the least. One approach, one that is increasingly necessary as we integrate and network processes and supply chains, is dynamic scheduling.

Rather than trying to control the process explicitly and calculate optimal schedules (knowing that perturbations will almost certainly occur), one can instead design the system to control itself. There are several, related approaches to implementing this. First, one can employ a market-based algorithm (Wellman 1993), e.g. individual machines that bid upon jobs according to some machine-related utility function such as how long it would take to complete a job of a particular type given its current configuration. The market among machines vying for jobs can be configured as an efficient market-clearing scheme without centralized control.

Consider the following manufacturing and scheduling problem that is the focus of this study: allocating trucks to paint booths (Morley & Schelberg 1993; Morley 1996; Morley & Ekberg 1998; Campos *et al.* 2001). Trucks roll off an assembly line at a rate of one per minute. While they are on the assembly line, they are allocated a color as desired by a customer, and at the end of the line, these trucks must be assigned to one of a small number of booths for painting. Ideally, we wish to assign trucks to booths currently configured for that color. If, for example, a booth is currently painting a truck red, then the best scenario is to assign it other trucks that need to be painted red. That way, a booth need not switch between colors, causing both a significant time delay and also a material cost: the wasted paint as the lines are flushed out and refilled with the new color.

Unfortunately, there a number of aspects that makes this process a challenge to schedule. First, customers' orders are unpredictable in time and content and there is a finite "information horizon" of the desired colors entering the system. Second, late arriving, high priority jobs, ones that need to "jump the queue," will negate any pre-determined optimal schedule. Third, booths are somewhat unreliable and may break down unexpectedly. Fourth and finally, booths have a queue in front of them where their assigned trucks wait for painting. However, this capacity is finite and very small. We cannot, for instance, assign all red trucks indefinitely to booth #1.

In as now classic study, Morley and colleagues (Morley & Schelberg 1993; Morley 1996; Morley & Ekberg 1998) demonstrated that this complex scheduling problem can be tackled dynamically with a market-based approach. In their scheme, the paint booths are the buyers, bidding against each other at auction for each truck that leaves the assembly line. According to their individual state (the color they are currently painting, the truck's priority, whether they are idle or broken etc.), the booths have different utility functions to the auction "lot" and place an appropriate bid using a small number of simple rules. When implemented in a real paint shop, this scheme worked extraordinarily well causing a dramatic increase in overall throughput and a significant reduction in wasted paint (*ibid.*).

Another similar approach to tackling this particular dynamic scheduling problem was developed independently by Campos *et al.* (2001) and Cicirello & Smith (2001, 2002; developed further by Nouyan 2002).

They adopted a swarm intelligence approach taking inspiration from the flexible, robust, and decentralized task allocation schemes employed by social insects such as ants and wasps (Bonabeau *et al.* 1999; Bonabeau & Meyer 2001). Here, the booths still bid against each other for trucks, but their bidding behavior is based upon the presumed response-threshold behavior of insects. A whole class of social insect response-threshold models has been developed in which a particular task, i , is associated with a task-specific stimulus, s_i . Individual workers, j , that tackle these tasks have task-specific thresholds, $\theta_{i,j}$, and their probability of response to task i , given the stimulus intensity and their individual threshold level, is often of the form $s_i^2 / (s_i^2 + \theta_{i,j}^2)$ (Bonabeau *et al.* 1999; Bonabeau & Theraulaz 1999). Thus, the greater a task's demand or the lower an individual's threshold, the more likely it is to tackle the task. These simple response processes were implemented in a simulation of the truck allocation process, and in many situations were shown to outperform Morley's scheme (Cicirello & Smith, 2001, 2002; Campos *et al.* 2001).

In this study, we simulate both Morley's market-based scheme and a response-threshold scheme (primarily focusing on Campos *et al.*'s (2001) simpler implementation). Our goal is to address a number of important issues:

First, how does one select the appropriate parameter values that maximize system-level performance? What, if any, are the tradeoffs between cycle time (the average duration to process a truck) and color switching rate (the number of paint flushes per truck)?

Second, how best does one choose among booths that have the same highest bid? In Morley's relatively discretized bidding system, there is potential for multiple booths to submit the same highest bids. We demonstrate that how the winner is decided in these situations—which we term the “breaking tie rule”—has a significant effect on the system's performance.

Third and finally, how best does one create specialist booths, ones that preferentially paint a single color? Threshold reinforcement is a feedback mechanism that is sometimes incorporated into threshold response models (e.g., Theraulaz *et al.* 1998; Bonabeau & Theraulaz 1999; Gautrais *et al.* 2001). That is, when an agent tackles task i , it decreases its threshold for task i but increases its thresholds for all other tasks ($j \neq i$), thus making it more likely to tackle task i in the future. In almost all response threshold models and applications, this updating process is “local”: an agent only adjusts its *own* thresholds. Unusually however, Campos *et al.* (2001) implemented a different process: a booth reducing its threshold for red, say, increases the red threshold for *all other* booths. Thus, the threshold reinforcement operates at a “global” scale with respect to the booths. Potentially, this increases the rate at which the system converges to efficient equilibrium behavior (E. Bonabeau, pers. comm.). But, is this really true? Thus, we investigate the convergence time and booth specialization for both local and global

threshold reinforcement. In the market-based scheme, the authors joke that they endow their paint booths with “sentient chicken brains” (meaning the simple rules; Morley 1996), and hence our title—a case of ants versus chickens.

2 Model

We model the truck painting process using stochastic simulation in discrete time. Customer-desired colors are assigned to the trucks while they are on the assembly line, and as each truck arrives at a “decision point” all booths that are operational bid for the truck using a bidding process described below. The truck is assigned to the winner and either moves into the booth, if it is currently empty, or join's the booth's queue (First In, First Out), which has a capacity of five. Painting takes three minutes but there is an additional time (and material) cost to the booth if it must change paint colors. This flushing of the paint lines and refilling with the appropriate color takes three minutes (Campos *et al.* 2001).

Our basic assumptions mirror those of Campos *et al.* (2001). We investigate two distributions of truck color. First, “uniform” in which a truck is assigned one of 20 colors randomly. Second, “skewed” in which 70% are assigned black, 15% white, 7% red, 4% blue, and random uniform distribution for the remaining 16 colors (Campos *et al.* 2001). As well as having an assigned color, each truck also has an assigned priority level (from $U\{1, \dots, 100\}$). When a truck enters the booth, there is a 5% chance that the booth will break down. This needs a uniformly random time $U\{1, \dots, 20\}$ to fix. Trucks in an inoperative booth's queue cannot be dispatched to other booths. Like Campos *et al.* (2001), we model a seven-hour shift with an arrival rate of one truck per minute, that is, a batch of 420 trucks, and eight identical paint booths.

Precise details of the market-based and ant-based bid functions are described in the following sections. Here, we describe the “resolution” process when two or more booths place the same, highest bid. First, we check for a match between the assigned color of the *last* truck in a booth's queue and the truck being bid for—a mismatch implies a color change and is thus unfavorable. If none or two or more of the contenders have a color match, then we select the booth with the shortest waiting time, that is the sum of the painting times plus color change time costs. Finally, if we are still unable to differentiate, a booth is chosen among the contenders randomly.

2.1 Market-based algorithm

In Morley's (1996; Morley & Schelberg 1993; Morley & Ekberg 1998) implementation, each paint booth is an agent that follows four simple rules:

- 1) Try to take another truck the same color as the current color;
- 2) Take particularly important jobs;
- 3) Take any job to stay busy;
- 4) Do not bid if paint booth is down or queue is full.

Unfortunately, Morley (*ibid.*) did not publish precise details of his bid functions, thus Campos *et al.* (2001), using the available information, proposed the following. If the booth is broken or it has no spare capacity in its queue, it does not bid. If no booth bids, the truck remains at the decision point. Otherwise, the booth's bid is

$$B_k = \frac{P \cdot w_i \cdot (1 + C \cdot c_{i,k})}{\Delta T^L} \quad (1)$$

where

$$\Delta T = q_k \cdot t_p + n_k^f \cdot t_f + t_r^k. \quad (2)$$

w_i is the priority of truck i . $c_{i,k}$ is an indicator variable taking the value 1 where there is a match between the color of the last truck in booth k 's queue and the truck i 's designated color, and 0 otherwise. ΔT is the total time before truck i would be painted, that is, the total expected time to paint the trucks currently assigned to booth k . It is the sum of the product of the painting time, $t_p = 3$ min., and the number of trucks q_k in booth k 's queue, plus the product of the paint flush time, $t_f = 3$ min., and the number of times that the next truck in line requests a different color than the truck in front of it, plus any time needed to complete painting a truck in the booth. P , C , and L are parameters to weight the relative importance of the three components, w_i , $c_{i,k}$, and ΔT .

2.2 Ant-based algorithm

Like Campos *et al.* (2001), we also implemented a threshold response based bidding process similar to that believed to occur in social insect division of labor (Theraulaz *et al.* 1998; Bonabeau & Theraulaz 1999; Bonabeau *et al.* 1999; Weidenmüller, in press). The booths are designed to operate in a similar manner to the basic threshold response scheme described in the introduction. Booth k 's probability of response to truck i , which has assigned color c_i , is defined as

$$P_k = \frac{D_{c_i}^2}{D_{c_i}^2 + \alpha \cdot \theta_{k,c_i}^2 + \Delta T^{2\beta}}. \quad (3)$$

D represents demand for a particular color, and is the equivalent of stimulus (see introduction). More specifically, D_j is a global demand for each color j calculated by summing the priorities of the unassigned trucks:

$$D_j = \sum_i w_i \cdot \delta(c_i - j), \quad (4)$$

where $\delta(\cdot)$ is Dirac function.

θ_{k,c_i} is a booth k 's threshold for color of truck i —each booth has one threshold for each color—and is bounded between θ_{\min} and θ_{\max} . ΔT is the same as in the market-based algorithm. α and β are parameters that weight the relative importance of demand terms and waiting time. Truck i will be assigned to the booth k , which has the highest P_k .

After truck i has been assigned to booth k the threshold values are updated. This occurs in one of two ways: locally and globally.

2.2.1 Local update

After booth k wins truck i , the booth decreases its own threshold to color c_i by an amount ξ (technically, by $\min\{\xi, \theta_{k,c_i} - \theta_{\min}\}$) and increases its threshold to all other colors by ϕ . [In the social insect literature, this is known as threshold reinforcement (Bonabeau & Theraulaz, 1999; Gautrais *et al.* 2001; Weidemüller, in press).] That is,

$$\theta_{k,c_i} \leftarrow \theta_{k,c_i} - \xi \quad (5)$$

and

$$\theta_{m,c_i} \leftarrow \theta_{m,c_i} + \phi. \quad (6)$$

This reinforcement mechanism thus has the potential to create booths that specialize in a color, thereby minimizing the need to switch among colors wasting time and resources flushing out the paint lines.

2.2.2 Global update

Whereas in local update, each booth only adjust its own threshold, in a global update rule, a booth adjust not only its threshold but those of all other booths. After booth k wins truck i , the booth decreases its response threshold to color c_i by ξ and increases the response threshold to color c_i of all other booths (denoted by m) by ϕ . That is,

$$\theta_{k,c_i} \leftarrow \theta_{k,c_i} - \xi \quad (7)$$

and

$$\theta_{k,j} \leftarrow \theta_{k,j} + \phi \quad \forall j \in S / \{c_i\}. \quad (8)$$

Here, one expects faster specialization because the difference between the thresholds, and hence the P_k bids, is exacerbated by this double positive feedback mechanism.

3 Results

3.1 Parameter space

To investigate the parameter value effects in both the market-based and ant-based algorithms, we ran 20 independent replicates for each set of parameters. We consider three reasonable minimands: 1) *number of color changes*, that is, the total number of color flushes per shift, i.e. for 420 trucks; 2) *average cycle time*, the average time a truck spends in the system, meaning leaving the assembly line (and arriving at the decision point, which may or may not involve a queue) to leaving a booth, and 3) *terminating time*, the time difference between the first truck leaving the assembly line and the last truck leaving a booth (a.k.a. *make span*). We explore the three parameters in market-based algorithm in the following ranges $P \in \{0, 10, \dots, 100\}$, $L \in \{0, 0.2, \dots, 5\}$, and $C \in \{0, 250, \dots, 10000\}$ and the six parameters in the ant-based algorithm as follows: $\alpha \in \{0, 10, \dots, 100\}$, $\beta \in \{0, 0.2, \dots, 5.0\}$, θ_{\min} and $\xi \in \{0, 1, \dots, 10\}$, and θ_{\max} and $\phi \in \{0, 1, \dots, 20\}$.

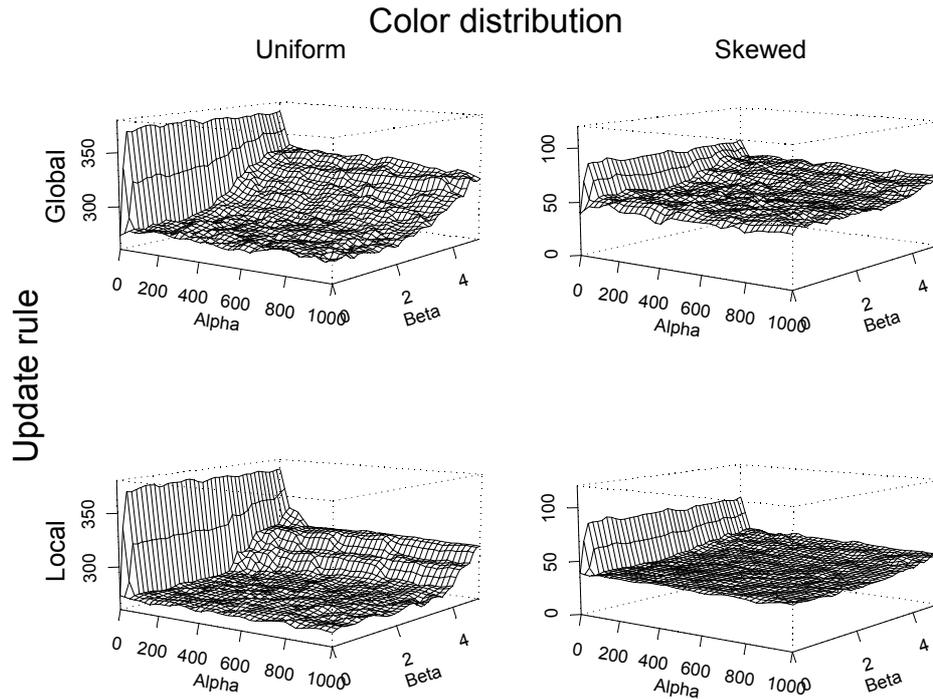


Figure 1: Number of color changes (z-axis) versus α (y-axis) and β (x-axis) for the ant-based algorithm. The two rows show results for the global and local update rules and the two columns, uniform and skewed color distribution.

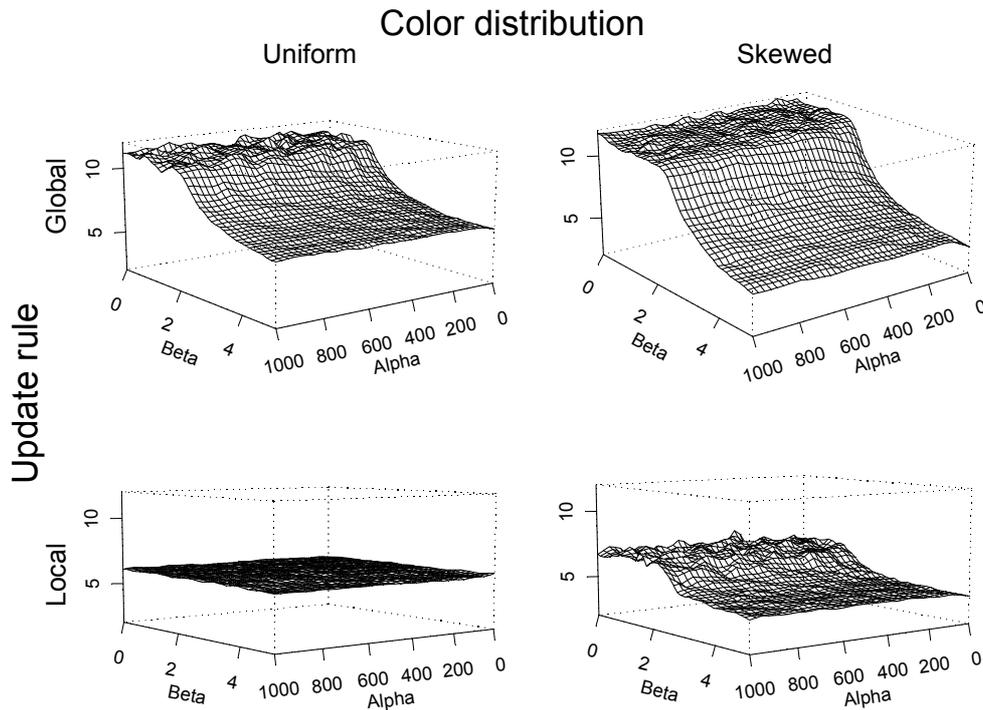


Figure 2: Average cycle time (z-axis) versus α (x-axis) and β (y-axis) for the ant-based algorithm for the two update rules and the two color distributions.

Figure 1 plots the number of color changes and figure 2 the average cycle time for the ant-based algorithm for the two update rules, global and local, and the two color distributions, uniform and skewed. The most striking feature is that in all cases the response surfaces are surprisingly smooth, often with broad, flat areas. Thus,

for instance, if we wished to minimize color changes then it is only necessary to set a low value of β —as can be seen in figure 1 this is true for both update rules and both color distributions. If on the other hand, we wish to minimize cycle time, we need only set a high value of β —again, true for both update rules and both color

distributions (Figure 2). The major implication is that for single objective functions, we do not need to resort to complex optimization techniques.

The second striking feature is that the system's behavior is highly insensitive to α (y-axis). Clearly, for color changes (Figure 1) there is a very large difference between $\alpha = 0$, which results in a very inefficient system with a high color change rate, and $\alpha > 0$, where the amount of color changing is much lower and, significantly, almost independent of the particular non-zero value of α . In contrast, average cycle time is minimized with $\alpha = 0$, and is higher for $\alpha > 0$ (Figure 2).

Finally, across parameter space, the skewed color distribution has a smaller range of cycle times but larger range of average cycle times than the uniform distribution. The extreme case is the cycle time for the local update rule (Figure 2), which exhibits negligible variation across α vs. β parameter space.

Figure 3 shows number of color changes for the market-based scheme explored across P (y-axis) vs. L (x-axis) parameter space for $C = 3,000$ (top row) and $C = 10,000$ (bottom row) for both uniform (left-column) and skewed color distribution (right-column). Figure 4 shows the average cycle times for the market-based scheme for the skewed color distribution for two values of C . It is clear that, qualitatively, the response surfaces are extremely similar to that of the ant-based scheme (Figures 1 and 2) and that P has an equivalent role and effect to α , and L is equivalent to β . The range of values is significantly larger than that of the ant-based scheme and,

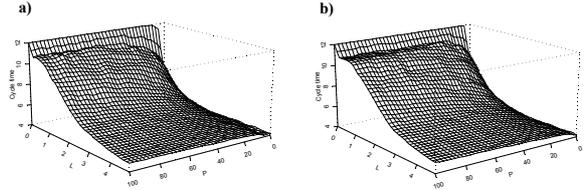


Figure 4: cycle time (z-axis) versus P (x-axis) and L (y-axis) for the market-based scheme for the skewed color distribution for a) $C = 5,000$ and b) $C = 10,000$.

in the main, higher—that is, less efficient—than the ant-based scheme (also found by Campos *et al.*, 2001 and Cicirello & Smith 2001, 2002).

From a more thorough exploration of parameter space in which we examined the proportion of a parameter's range that causes at least a 10% change in system performance, we find that the dimensions of parameter space can be reduced. We focus on the ant-based scheme because it outcompetes the market-based scheme in most situations (Campos *et al.* 2001, Cicirello & Smith, 2001, 2002, and our results) and find that the parameters can be divided into three categories:

- **Insignificant:** θ_{\min} and ξ are parameters that are essentially unnecessary and may be set to some arbitrary constant. They account for less than 20% of the range in system performance across parameter space.
- **Moderate:** α , ϕ , and θ_{\max} are parameters that all have a moderate effect upon system performance, that is, accounting for 20–60% of the range.

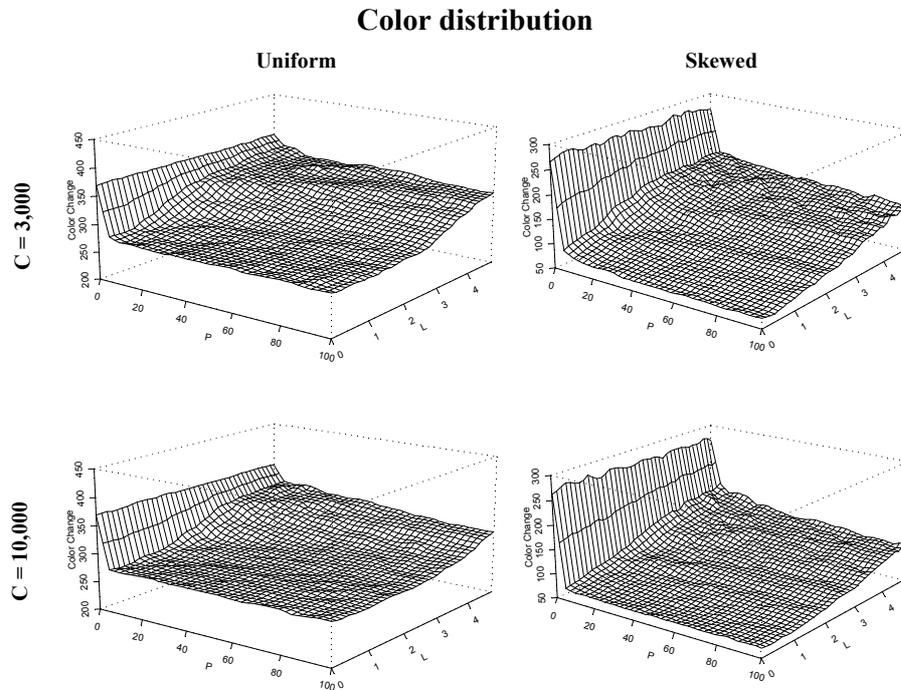


Figure 3: number of color changes for the market-based scheme for uniform (left-hand column) and skewed (right-hand column) color distribution for $C = 3,000$ (top row) and $C = 10,000$ (bottom row). For each graph, color changes appear on the z-axis and P and L on the y- and x-axes respectively.

- **Sensitive:** figures 1 and 2 clearly demonstrate that the system’s results are highly sensitive to β , the key parameter of the system.

3.2 Mixed objective functions

The waiting time parameter, β , plays a key role in determining the system’s performance. However, as hinted in the previous subsection, there is a tradeoff between the minimands; for instance, high β causes low cycle times but more frequent color changes. Figure 5 illustrates the tradeoff between the three minimands more explicitly as a function of β . We have already established that the results are fairly insensitive to α (Figure 1), thus we average the curves across non-zero α . These are then normalized by subtracting the minimum of each distribution (combining both local and global) and dividing by the maximum range of all four cases. This is repeated for the three minimands as a function of β .

First, it clear that there is a very strong positive correlation between terminating time and cycle time. It is for this reason that we did not discuss terminating time in the previous subsection—in brief, the same results hold for both terminating time and cycle time. Figure 5 clearly shows a tradeoff between rate of color changes and both terminating time and cycle time (with perhaps the exception of local update rule and uniform color distribution). This tradeoff makes sense intuitively: we can minimize color changes by preventing trucks being assigned to booths that would have to change color but this comes at the expense of unassigned trucks waiting at the decision points, shorter lines in front of the booths,

and, at times, even idle booths, all of which contribute to slower throughput.

Figure 5 may even be of practical use. The tradeoff can be changed as desired by varying just a single parameter β . Moreover, clearly there is a real cost associated with wasted paint and reduced throughput. Thus, figure 5 can be used to develop a plant- or line-specific weighted mixed objective function. For instance, suppose return on investment is somehow maximized by minimizing the sum of the factors multiplied by their costs, say c_{color} , $c_{\text{terminating}}$, and c_{cycle} . One can very easily find β that minimizes $\text{color change rate} \times c_{\text{color}} + \text{terminating time} \times c_{\text{terminating}} + \text{cycle time} \times c_{\text{color}}$, without having to resort to more complex optimization techniques.

3.3 Advantages of dynamic scheduling

In a fluctuating job shop environment, dynamic scheduling algorithms often have an advantage over more traditional scheduling schemes as they are flexible and robust, designed to cope with perturbations and other noise (Morley 1996). Moreover, our results and those of Campos *et al.* (2001), Cicirello & Smith (2001) and Nouyan (2002) all demonstrate that ant-based algorithm usually outperforms the market-based algorithm. The obvious question is why this is so. In this subsection, we examine three characteristics of job shop scheduling that may contribute to this superior performance, namely agent specialization, steady state dynamics and the breaking tie rule. We use the parameter values given in Campos *et al.* (2001).

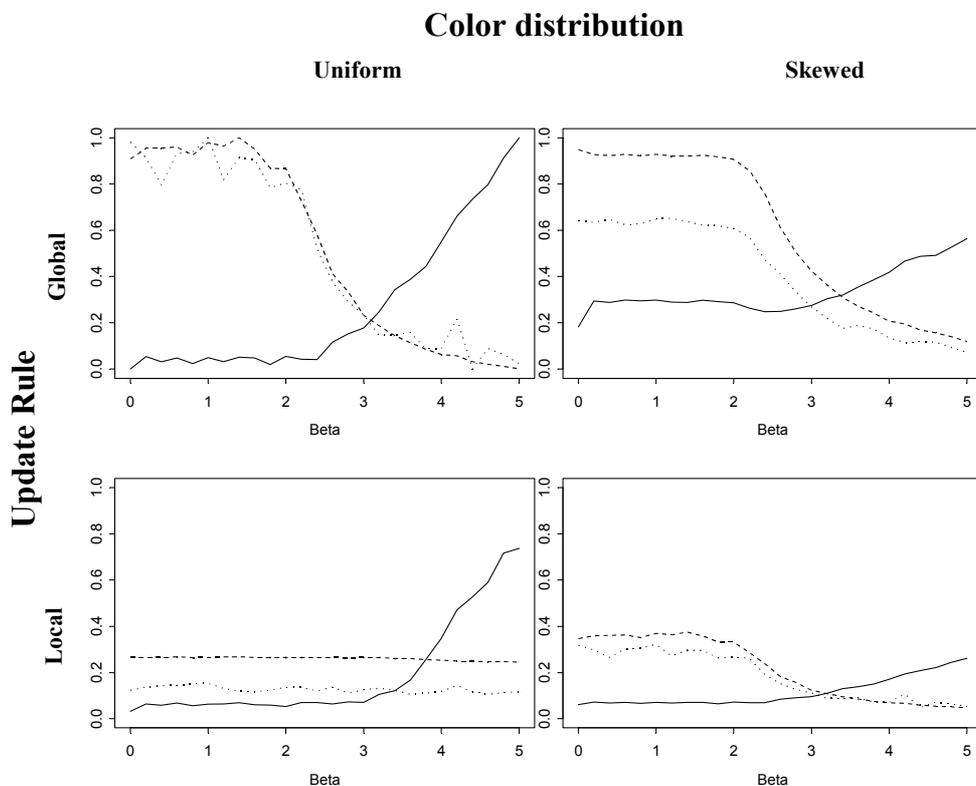


Figure 5: Normalized color changes (solid line), terminating time (dotted line) and average cycle time (dashed line) for uniform (left-hand column) and skewed color distribution (right-hand column) for global and local update rule (top and bottom row respectively) for the ant-based scheme. (See text for details of normalization process).

3.3.1 Agent specialization

Our hypothesis is that the booths in the ant-based scheme are greater color specialists than those of the market-based scheme. To test this, we used two measures of worker specialization. The first is Gautrais *et al.*'s (2002) specialization metric. Essentially, this measures the rate at which a booth changes color (hence is the inverse of color block size). Let T_j be the number of transitions to a different color divided the number of transitions between trucks for booth j . Thus, with R and B representing red and black trucks respectively, the sequence RRRBBBRR has 2 color transitions (underlined) and 8 truck transitions, therefore resulting in $T = 2 / 8 = 0.25$.

Our second metric is social entropy (Shannon 1948; O'Donnell & Jeanne, 1990). This metric defines the entropy of a booth j as

$$E_j = -\sum_i p_{ij} \times \ln(p_{ij}) \quad (9)$$

where p_{ij} is proportion of trucks that booth j painted that were of color i . Pure generalists in which each color is painted equally (that $p_{ij} = 1 / \text{total number of colors}$; $1 / 20$ in this study) minimizes E_j (i.e., $\ln(n^n)$ for n colors) whereas pure specialists in which $p_{ij} = 1$ for one color and zero for all other colors maximizes E_j (i.e., ∞). In short, higher values represent higher degree of specialization.

For each simulation, we calculated the two specialization scores for each booth and took the two median values among the 8 booths. This was replicated 100 times (to give 100 medians), and was repeated for all three schemes: ant-based with global update, ant-based

with local update, and market-based. Each set of 100 medians was normally distributed thus we used one-way ANOVA with post-hoc pairwise comparisons.

We found that mean median entropy differed significantly among the three algorithms ($F = 178.11$, d.f. = 299, $p \approx 0$): global (entropy score \pm s.d. = 0.5934 ± 0.193) > local (0.4840 ± 0.207) > market-based (0.9280 ± 0.099). Mean median specialization also differed significantly ($F = 67.92$, d.f. = 299, $p \approx 0$) but not between global and update rule (0.12986 ± 0.0439 vs. 0.12437 ± 0.0397 respectively) but both of these did differ significantly at the 95% significance level from the market-based scheme (0.18088 ± 0.0281). In short, we confirmed our original hypothesis: the market-based scheme has a lower degree of color specialization among its booths thus resulting in a smaller color block size. However, differences between global and local updates are less clear-cut.

3.3.2 Steady-state dynamics

As mentioned earlier, Campos *et al.*'s (2001) study differs from the vast majority of other response-threshold based models as it uses a global rather than local threshold update rule. A priori, there are two possible hypotheses why a global update rule might be favored. First, it may have a superior performance at equilibrium than local update; second, the system may reach equilibrium faster. The intuition behind these two hypotheses is that the global update rule adjusts the response thresholds of every agent for a particular color throughout the system and the adjustments happen in two directions—a

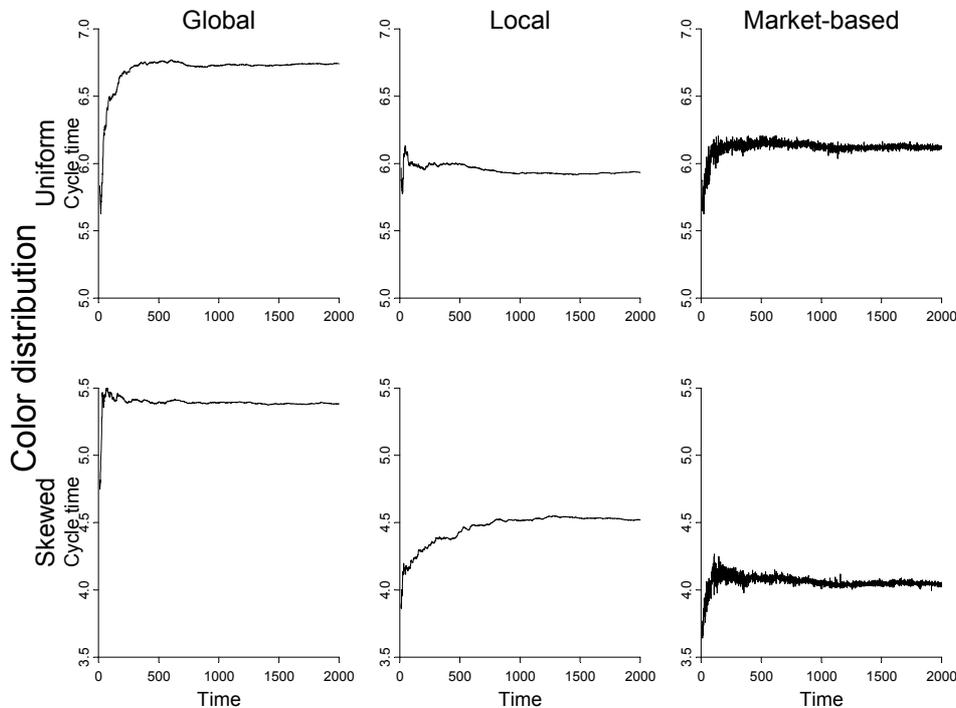


Figure 6: average cycle (since $t = 10$ minutes) versus time for uniform (top row) and skewed (bottom row) color distribution for ant-based global update rule (left-hand column), ant-based local update scheme (middle column) and market-based scheme (right-hand column).

threshold reduction for the booth to which the truck is assigned and a threshold increase to all other booths for that color.

Figures 1 and 5 demonstrate that we can reject our first hypothesis: the global update rule does not provide as good system performance as the local update rule. We tentatively suggest that the global update rule allows some booths to specialize quickly. Moreover, it is possible that with a global update rule, a booth could specialize in more than one color, or at least have multiple low response thresholds, as this rule lacks the interdependence present in the local update rule—that as a booth’s threshold for one color decreases, its thresholds for the other colors increase.

To test the second hypothesis, that the system reaches equilibrium more quickly, we significantly increase the number of trucks and stop the simulation after 2,000 simulated minutes. Here we focus on average cycle time to date, meaning the average cycle from time $t = 10$ to the current time point (in effect, a cumulative average) and system utilization.

Figure 6 shows that the two ant-based schemes are smoother than the market-based scheme. More importantly, we can see that there is no significant difference among the three algorithms in time taken to reach dynamic equilibrium for the uniform color distribution ($t \gg 500$) while for the skewed color distribution, the three schemes differ: global update rule ($t = 250$) settles faster than market-based ($t = 400$) which in turn settle far more quickly than the local update rule ($t = 1,000$). Although figures 1 and 6 show that cycle time and number of color changes for the local update rule is lower than the global update rule, when we consider all parameter combinations, however, systems with a global update rule reach the steady state more quickly and within a less variable time frame. Therefore, we conclude that a global update may be more advantageous than local update.

3.3.3 Breaking-tie rule

One drawback of these dynamic scheduling schemes is the potential for multiple booths to submit the same bid. This may arise for several reasons: first, booths may have the same initial configuration; second, the color of the truck the booths are bidding for is a discrete variable; and finally, a booth’s state (busy/idle/broken), size of its queue, and the paint color in its paint lines are also discrete. [For the market-based scheme, G. Ekberg (pers. comm.) suggests the bidding function should capture how far a truck physically travels from the decision point to the booth, one aspect that will distinguish among booths.]

Campos *et al.* (2001) suggests a reasonable way to overcome this problem of breaking ties by comparing the waiting time and the colors in the queue among the highest-bid placing booths. This additional information and information-gathering processes might be expensive and affect the robustness of the dynamic scheduling. Thus, it is important to examine how breaking ties affects the system’s efficiency.

In Section 2, we described the hierarchy of decisions used to decide the winner in the market-based scheme when there are multiple booths with the same highest bid:

- 1) Color of last truck in each booth’s queue;
- 2) Booth with shortest waiting time;
- 3) Choose randomly.

At each level, we used the criterion to test whether that distinguished among contenders. If not, then we moved to the next level down. Here, we implement variants of these rules: “Regular” in which we follow the normal hierarchical scheme above, “Color match only” in which we do not incorporate criterion 2 (waiting time), “Waiting time only” in which we do not incorporate criterion 1 (color), and “Random,” where we ignore both criteria 1 and 2 and simply select a winner at random.

Figure 7 shows the results of the various breaking tie rules on the number of color changes and average cycle time in a single shift, i.e. 420 trucks. We can see that the

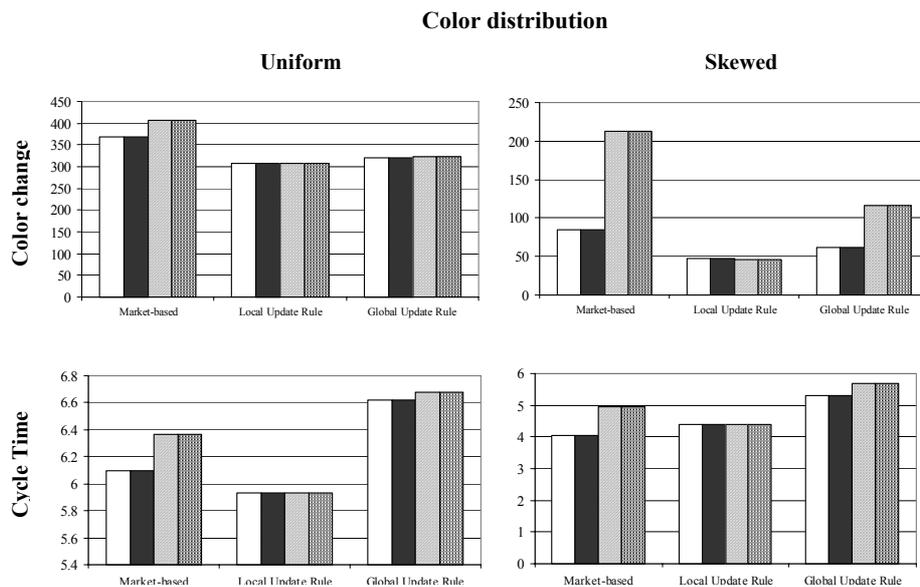


Figure 7: Number of color changes (top row) and average cycle time (bottom row) for the uniform (left-hand column) and skewed (right-hand column) color distribution for several breaking tie rule variants: market-based (white), color only (black), waiting time only (hatched), and random (plaid) (see text for details). The three groups of bars are market-based, ant-based with local update rule and ant-based with global update rule respectively.

waiting time only produces the same result as random, and color match only is equivalent to regular, regardless of the scheme or color distribution. Hence, color is the dominant factor and waiting time can be eliminated entirely from the breaking tie rule. Surprisingly, the performances of the local update rule with and without the breaking tie are close. Practically, this makes the local breaking tie rule more preferable because it is insensitive to both initial condition and breaking tie rule. These intriguing, and somewhat surprising, results can be explained as follows: the key is the response threshold mechanism. The market-based scheme doesn't have a response threshold and the global update rule has higher "forgetting rate" than local update. Therefore, one can view the response threshold as a memory and the past assigned sequence as experience. When a new truck enters the decision point, each booth perceives it differently based on their experience and memory so they place the bid differently.

4 Discussion

Our study has revealed a number of important insights about these two decentralized dynamic scheduling schemes. Whereas Campos *et al.* (2001) used a genetic algorithm to parameterize their system, we find that the "response surfaces" are surprisingly smooth and relatively insensitive to the parameter values. For a given objective, e.g., throughput, there are surprisingly broad regions of parameter space with which it can be achieved. These results are particularly useful as we identify the key, most sensitive parameters, some that are relatively insensitive, and several that are effectively irrelevant and can be ignored. Our analysis of local versus global threshold update too may be of particular broader significance and importance given its frequent use in swarm intelligence applications.

Acknowledgments

We thank Gregg Ekberg, Eric Bonabeau, and Shervin Nouyan for useful discussion, details, and advice. This study was generously supported by funds from Georgia Institute of Technology's Institute for Sustainable Technology and Development. We thank the director, Prof. Bert Bras, for both his financial and moral support.

Bibliography

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

Bonabeau, E. & Meyer, C. (2001) Swarm Intelligence: a whole new way to think about business, *Harv. Bus. Rev.* (May), 107–114.

Bonabeau, E. & Theraulaz, G. (1999) Role and variability of response thresholds in the regulation of division of labor in insect societies. In: *Information Processing in*

Social Insects (C. Detrain, J.L. Deneubourg and J.M. Pasteels, Eds.), Birkhäuser, Basel, 141–163.

Campos, M., Bonabeau, E., Theraulaz, G. & Deneubourg, J.L. (2001) Dynamic scheduling and division of labor in social insects. *Adaptiv. Behav.* **8**: 83–92.

Cicirello, V.A. & Smith, S.F. (2001). Wasp-like agents for distributed factory coordination. Technical Report CMU-RI-TR-01-39, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 2001.

Cicirello, C.A. & Smith, S.F. (2002). Distributed coordination of resources via wasp-like agents. In: *The First Goddard/JPL Workshop on Radical Agent Concepts (WRAC'02)*, NASA.

Gautrais, J., Theraulaz, G., Deneubourg, J.L. & Anderson, C. (2002). Emergent polyethism, as a consequence of increased colony size in insect societies. *J. theor. Biol.* **215**: 363–373.

Morley, R. (1996) Painting trucks at General Motors: the effectiveness of a complexity-based approach. In: *Embracing Complexity: Exploring the Application of Complex Adaptive Systems to Business*. Ernst & Young Center for Business Innovation, Boston, 53–58

Morley, R. & Ekberg, G. (1998) Self-organizing military logistics. In: *Embracing Complexity: A Colloquium on the Application of Complex Adaptive Systems to Business*, The Ernst and Young Center for Business Innovation, Cambridge, 97–102.

Morley, R. & Schelberg, C. (1993) An analysis of a plant-specific dynamic scheduler. In: *Final Report, Intelligent, Dynamic Scheduling for Manufacturing Systems*, 115–122.

Nouyan, S. (2002) Agent-based approach to dynamic task allocation. In: *Ant Algorithms* (M. Dorigo, G. Di Caro, and M. Sampels, ed.s), *Lect. Notes Comp. Sci.* **2463**, 28–39.

O'Donnell, S. & Jeanne, R.L. (1990). Forager specialization and the control of nest repair in *Polybia occidentalis* Olivier (Hymenoptera: Vespidae). *Behav. Ecol. Sociobiol.* **27**: 359–364.

Shannon, C.E. (1948) A mathematical theory of communication. *The Bell System Technical J.* **27**: 379–423 & 623–656.

Smith, A. (1776) *Wealth of Nations* (reprinted 1982). Penguin Books, London.

Taylor, W.F. (1911) *Principles of Scientific Management*. Harper Bros., New York.

Theraulaz, G., Bonabeau, E. & Deneubourg, J.L. (1998). Response threshold reinforcement and division of labour in insect societies. *Proc. R. Soc. Lond. B* **265**: 327–332.

Weidenmüller, A. The control of nest climate in bumble bee (*Bombus terrestris*) colonies: interindividual variability and self-reinforcement in fanning response *Behav. Ecol.*, in press.

Wellman, M.P. (1993) A market-oriented programming environment and its application to distributed multicommodity flow problems. *J. Artif. Intel. Res.* **1**: 1–23.