Scheduling in a Multi-Processor Environment with Deteriorating Job Processing Times and Decreasing Values: the Case of Forest Fires

Costas P. Pappis, Nikos P. Rachaniotis
University of Piraeus, Department of Industrial Management and Technology, 80 Karaoli & Dimitriou str., 185 34, Piraeus, Greece, {nraxan, pappis}@unipi.gr

In forest fire fighting, time and effort required to control a fire increase if fire containment effort is delayed. The problem of scheduling multiple resources employed as parallel identical or non-identical processors in order to contain N≥2 fires may be tackled using the concept of deteriorating jobs. In this paper, the above problem is stated and a model is formulated, the criterion being to maximize the total remaining value of the burnt areas and a heuristic algorithm to tackle this NP-hard problem is proposed.

Keywords: Forest fires, scheduling, parallel processors, deteriorating jobs, multiprocessor jobs

1. Introduction

Job scheduling research has been recently extended to situations with multiple parallel identical machines (processors) and deteriorating (increasing) processing times, depending on the jobs' processing starting times. This situation appears in many real applications, e.g. scheduling industrial processors whose productivity reduces over time, scheduling industrial maintenance, repair and cleaning assignments, scheduling steel rolling mills and metallurgical processes, repayment of multiple loans, etc.

In (Chen, 1996) and (Chen, 1997) it has been shown that the parallel identical machines problem with simple linear deterioration processing times, under the criterion of minimizing the sum of completion times, is NP-hard in the ordinary sense. In (Mosheiov, 1995) heuristics were proposed for the parallel identical machines problem with linearly or stepwise deteriorating functions. In (Mosheiov, 1998) a problem with parallel identical machines and linearly deteriorating processing times was studied under the makespan minimization criterion. This problem was shown to be NP-hard and a heuristic was proposed. In (Hsieh and Bricker, 1997) three heuristics were proposed for the parallel identical machines problem with linearly decreasing processing times under the makespan minimization criterion.

Another type of problems in the area of machine scheduling is when the jobs' value is a deteriorating (decreasing) function of time, with constant processing times. The value can decrease due to several reasons like customer dissatisfaction due to delays, seasonal nature of the product, technological devaluation and/or physical degradation and lack of inventory capacity ((Smith, 1956), (Lenstra et al., 1977), (Fisher and Krieger, 1984), (Buyukkoc et al., 1985), (Kawaguchi and Kyan, 1986), (Alidaee, 1993), (Voutsinas and Pappis, 2002)).

A new version of the single-machine problem is to find the optimal scheduling policy if the processing times deteriorate, i.e., increase depending on the processing starting time, and the jobs' values also deteriorate, i.e. decrease over time. The scheduling criterion is the maximization of the total jobs' remaining value at the time that processing of all jobs is completed. Different jobs' processing times and value functions can be considered and some easy to prove properties linking this criterion with the well known minimization criterion of the sum of completion times and weighted completion times have been derived (Rachaniotis, 2004).

Regarding the above-described problem, in (Teunter and Flapper, 2003) a production-rework environment is examined. It is assumed that rework processing time and costs per item increase linearly with the time span that an item is held in stock and awaits rework. Defects occur stochastically, and the objective is to determine the production lot size in such a way that the expected system profit be maximized. In (Inderfurth et al., 2005) a deterministic lot-size model was presented, focusing on a manufacturing system within which production as well as rework activities are carried out. Each changeover from production to rework and vice versa causes a fixed set-up cost and set-up time. Due to the deterioration caused by waiting, the cost and time of rework operations increase linearly. Considering set-up and inventory holding costs as well as set-up times, optimisation algorithms were developed in order to minimize the total cost.

Finally, a relevant application of this problem, which is tackled in (Rachaniotis and Pappis, 2006), is scheduling a single fire fighting resource when there are several forest fires to be suppressed. The time required to suppress a forest fire increases if there is a delay in the beginning of the containment's effort, while the value (monetary or of some other kind) of the affected area is assumed to decrease quadratically with the duration of the containment effort. Consequently, scheduling the available fire fighting resource in the case of multiple forest fires may be treated as a job-scheduling problem with deteriorating processing times, with the objective of maximizing the total remaining value of the burnt areas after the completion of the containment operation. The times needed for the resource to travel from one site to another were considered as set-up times (as they would be in a typical manufacturing environment). A solution methodology was proposed with water as fire suppression means. A specific model for the fire rate of spread was used and an equation for the 'processing time', i.e. the time needed for fire containment, was derived, which was related to the areas' decreasing value.

In this paper, the extension of the previously described problem of scheduling $n>1$ jobs (forest fires) on $m>1$ non-identical, parallel processors (fire fighting resources) will be examined. This is one of the main problems of the forest fire fighting operational centers in Greece and all over the world and the authors' research team has established a close co-operation with the Greek Fire Corps in the past few years. The above problem with multiprocessor jobs is stated and a model is formulated, the criterion being to maximize the total remaining value of the burnt areas and a heuristic algorithm to tackle this NP-hard problem is proposed. Multiprocessor jobs require more than one processor at the same moment of time. This is not the classical scheduling theory's usual assumption that a job is executed on one processor at a time (Drozdowski, 1996b).

2. Statement of the problem

In forest fire fighting practice, a forest region is normally assigned to several fire-fighting resources ('processors' in the sequel), which can be identical or not, e.g. water-carrying fire engines, crews with hoses connected to waterspouts, and water or retardant dropping aircraft and helicopters. A variation of Maximal Covering Location Model (MCLM), where information regarding the risk of fire ignition is embedded, may be utilized to find the proper location of the available processors in such a way as to cover optimally the considered forest region (Dimopoulou and Giannikos, 2004).

In this model, the forest region is subdivided into non-uniform, space exhausting and non-overlapping (demand) areas that must be covered. An initial value is assigned to each area, which decreases over time when a fire has ignited. This value can be monetary or ecological or of some other kinds. The value in each area decreases at a different rate, depending on the fuel type, the ecological value, whether it is inhabited or not, etc, requiring a different level of priority in a fires' suppression sequence (Rachaniotis and Pappis, 2006).

If the fire's fuel bed surface has a uniform slope and the fuel and meteorological conditions are homogeneous, then empirical evidence shows that a forest fire will expand with variable shapes, which are often approximated by ellipses with various length-to-breadth ratios. This will be the shape of the (expanding over time) fire's perimeter used in this paper. The major axis of the perimeter lies along with the wind direction. Higher wind velocities give greater Rates of Spread (ROS) and higher length-to-breadth ratios of the perimeter shape. If the

surface is sloped then the major axis of the perimeter moves upslope from that of the wind direction and lies along to what is known as "the effective wind direction" (Richards, 1999).
The primary goal of a fire containment effort is the earliest possible containment of the fire, which generally means the desperation of compatible fuel and oxygen at the fire's perimeter. This is most commonly achieved by a rapid encirclement of the fire with a fire strip, whose width depends on the intensity of the fire. In this paper this is assumed to be constant.
If more than one fire ignite either simultaneously or consecutively, it is quite possible that not all of them can be serviced as soon as they are reported. Then the decision maker must make a quick preliminary assessment of initial attack requirements and schedule each fire to a position in a priority initial attack queue. Most often, many suppression resource units are allocated to many fires when they are reported. This initial attack system may be modelled using simulation techniques as a multiserver queuing system with time dependent servicing times (Martell et al., 1984). In (Fried and Fried, 1996), a containment algorithm facilitates fire simulation with different rates of spread, tactics and resources arrivals. The outputs are the final fires' size, the containment time and the number of the resources to be utilized, provided that the fire does not escape initial attack.
The processing time, i.e. the time needed for a fire's suppression, increases (deteriorates) at a rate depending on the time elapsed from the ignition of the fire, the kind of fuel in the area that the fire burns (different fires' ROS) and the processors' fire containment efficiency. Consequently, since time and effort required to control a forest fire increases with time, scheduling the available fire fighting resources in the case of multiple forest fires may be treated as a job-scheduling problem with deteriorating job processing times and decreasing job values. The objective is the minimization of the total damage caused by the fires to the burnt areas or, equivalently, the maximization of the total remaining values of the areas.
This situation involves some special characteristics, including the following:

a. The parallel non-identical processors are unrelated and their containment rate is job-dependent (fire-dependent).

b. Fires can be considered as multiprocessor jobs, i.e. more than one processor is required at the same time for a specific fire.

c. Pre-emption, i.e. the situation where the processor is called to suppress a fire while it is busy suppressing another one, is allowed.

d. Jobs have variable profiles, i.e. the number of processors employed for specific fire suppression can change during the containment effort.

2.1 Notation

Let:
F: be a set of N forest fires $\{F_1,\ldots,F_N\}$ with $L \leq N$ different values of ignition times ("release times"), $0=r_1<r_2<\ldots<r_L$.
$V_i(t)$: be the remaining value of the area that F_i burns at time t.
V_{i0}: be the initial value of the area that F_i burns.
R: be the set of fire fighting kinds of resources $\{R_1,\ldots,R_K\}$, i.e. it is assumed that there are K different subsets (kinds) of non-identical resources with M_j identical processors of each kind, j=1, 2,…,K, and $\sum_{j=1}^{K} M_j = M$, i.e. M is the total number of the parallel unrelated processors (identical or non identical).
$t_0>0$: be the time instance of the beginning of the forest fire suppression effort.
$\rho_{ij}(t)$: be the containment rate of F_i by resource of kind j at time t.
$\underline{s}_i(t) = (s_{i1}(t),\ldots,s_{iK}(t))$: be the vector of the numbers of resources employed for the suppression of F_i at time t, $i=1,2,\ldots,G \leq N$.
$P(\underline{s}_i(t),t)$: be the processing time, i.e. the time needed to contain F_i at time t.
$Area_i(t)$: be the area of the elliptical fireline strip that has to be created in order to contain F_i at time t.

$\alpha,\beta,\gamma,\varepsilon$: be constant parameters depending on the kind of the area that F_i burns (the fires' ROS differs in different demand areas) and the meteorological conditions.
$t_{max,i}$: The "containment escape time limit" of F_i, i=1,2,…,N. F_i has escaped initial attack if its containment time exceeds $t_{max,i}$. It can be calculated at the ignition time of F_i.
$\underline{\delta}_i = (\delta_{i1},...,\delta_{iK}): \quad P(\underline{\delta}_i, r_i) = t_{\max,i}, with\ \delta_{ij} \le M_j$
be the vector of "minimal requirements" for the suppression of F_i, i.e. the necessary resources in order to contain F_i in $t_{max,i}$ and not to consider it as an escaped initial attack fire.
$h_i(t)$: be the shortest time required to suppress F_i at any time t. In this paper it will be called the ***height*** of F_i at time t.
$C(\underline{s}_i(t)) : h_i(C(\underline{s}_i(t)) = 0$ be the completion time of F_i's suppression

2.2 Assumptions

a. The available processors fully cover the region examined and are located inside it.
b. The processors can always reach any fire.
c. The fuel and meteorological conditions remain constant (homogeneous conditions) at each demand area.
d. The burn time of an area depends on its fuel and is not affected by the fuel of adjacent areas.
e. The value of the burning area that F_i burns deteriorates quadratically over time with a constant rate w_i, i.e.

$$V_i(t)=V_{i0}-w_i C^2(\underline{s}_i(t))$$

f. In every fire containment effort there is a number of time intervals where the assignment of fires to resources is constant.
g. Different resource types can substitute one another.

3. The model

Fireline is rarely built at a constant rate for the duration of an initial attack effort. The productivity decreases as resources' traveling times ("set-up times" in a typical job-scheduling problem in a manufacturing environment) are included, crews' fatigue appears and water and retardant supplies are depleted. This is called 'drop-off' phenomenon (Fried and Fried, 1996). The result is a stepwise containment rate function:

$$\rho_{ij}(t) = \begin{cases} a_{ij}, t \le t_{ij} \\ b_{ij}, t > t_{ij} \end{cases}, b_{ij} < a_{ij}$$

where t_{ij} change as the containment effort "clock time" increases.
The processing time equation is ($Area_i(t)$ is calculated in Rachaniotis and Pappis, 2006):

$$P(\underline{s}_i(t),t) = \frac{Area_i(t)}{\sum_{j=1}^{K} s_{ij}(t)\rho_{ij}(t)} = \frac{\alpha\beta^2 t^{2\gamma} + 2\alpha\beta\varepsilon t^{\gamma+1} + \alpha\varepsilon t^2}{\sum_{j=1}^{K} s_{ij}(t)\rho_{ij}(t)}$$

The height of F_i at time t (defined as the area of the elliptical fire fighting strip that is not contained at time t divided by the maximum containment rate) is calculated using the equation

$$h_i(t) = \frac{Area_i(t) - \int_{t_0}^{t} P(\underline{s}_i(\tau),\tau)\sum_{j=1}^{K} s_{ij}(\tau)\rho_{ij}(\tau)d\tau}{\sum_{j=1}^{K} \mathrm{M}_j \rho_{ij}(t)}$$

The initial height of F_i (for $t=t_0$) is

$$h_i(t_0) = \frac{Area_i(t_0)}{\sum_{j=1}^{K} \mathrm{M}_j \rho_{ij}(t_0)}$$

The objective is to minimize the total damage caused in the burned areas,

$$\min \sum_{i=1}^{N} w_i C^2(\underline{s}_i(t))$$

or equivalently to maximize their total remaining value:

$$\max \sum_{i=1}^{N} V_i(t)$$

4. Heuristic Algorithm

4.1 Algorithm's description

The heuristic presented here for tackling the aforementioned problem uses some ideas of Drozdowski's algorithm. (Drozdowski, 1996a) examined a problem of scheduling parallel applications in a multiprogrammed multiprocessor system. The preemptive case was addressed with a variable number of processors used by a task over time. A low-order polynomial time algorithm was proposed for minimizing the makespan, which schedules tasks according to their "level" which is the time required to finish them.

The intervals between the fires' ignition times are considered consecutively. Subintervals are created where the ratios $w_i/h_i(t)$ are not changing. The rational for using these ratios stems from the well known Smith's rule for solving optimally the problem $N/1/\min\Sigma a_i C_i$ with constant processing times P_i, which is to schedule jobs in non-decreasing order of P_i/a_i (Smith, 1956). This heuristic is also used in the problem $N/1/\min\Sigma a_i C_i^2$ with constant processing times, which is a much more simple but quite similar to the problem examined in this paper ((Della Croce et al., 1995), (Mondal and Sen, 2000)).

Fire suppressions are assigned to processors, and fires with higher ratios are given preference. When there are more processors available than those, which are simultaneously required by the fires already ignited, a maximal possible number of processors are assigned. Otherwise, processors are shared by fires with equal ratios so that their heights decrease at the same rate. The time length of the current assignment is calculated. This assignment of processors changes in three cases: a) The height for some fire with initially higher ratio becomes equal to the height of some fire with initially lower ratio and the processing times must be recalculated in order to obtain the same rate of height decrease for equal ratios fires, b) The fire suppression with the shortest completion time in the release times' interval finishes or c) The end of the interval is encountered and fire suppression tasks in the next interval must be considered. Finally, McNaughton's wrap-around rule (McNaughton, 1959) is used to schedule pieces of fire suppression tasks. This rule uses a new processor for containing a fire only if the previous processor is fully employed for the examined time interval.

Note that no information about fire ignitions in the next release times' intervals is necessary to schedule the suppression of fires with release times less than r_{k+1}. Hence, the above algorithm is a real-time (dynamic) algorithm and it is synchronous (can be run on-line), meaning that it builds sub-optimal schedules using only the information about fires that have already ignited. The algorithm's validity proof is similar to the one in (Drozdowski, 1996).

A schematic overview of the algorithm, which is analytically presented in the next sub-section, is the following:

Step1: Schedule fires in non-ascending order of ratios $w_i/h_i(t)$.

Step2: Assign resources to fires already ignited.

Step3: Check for any needs in altering resources' assignments.

Step4: Use McNaughton's rule for scheduling pieces of fire suppression tasks.

Step5: Calculate new fires' heights.

4.2 The algorithm

```
t=t0
for k=1 to L do
   Setk=set of fires with ignition time rk
```

Schedule fires in Set_k in non-ascending order of ratios $w_i/h_i(t)$. Then the fires are denoted $F_{[1]}, F_{[2]}, \ldots$

while ($r_{k+1}>t$) and ($\exists\, F_i \in Set_k$: $h_i(t)>0$) do

(*procedure of assigning resources*)

$\underline{s}(t) = \underline{0}$

$avail_j=M_j$ ($\Sigma M_j=M$) (*available processors of kind j *)

while ($avail_j>0$ for some kind j) and $|Set_k|>0$ do

if $\sum_{F_{[i]}\in Set_k} \delta_{[i]j} > avail_j$ then

for i=1 to $|Set_k|$ do

for j=1 to K do

$s_{[i]j}(t)=\delta_{[i]j} \cdot (avail_j$ with highest containment rates$)/ \sum_{F_{[i]}\in Set_k} \delta_{[i]j}$

$avail_j=0$

else

for i=1 to $|Set_k|$ do

for j=1 to K do

$s_{[i]j}(t)=\delta_{[i]j}$ with highest containment rates

$$\text{avail}_j = \text{avail}_j - \sum_{i=1}^{|Set_k|}\sum_{j=1}^{K}\delta_{[i]j}$$

(* end of procedure*)

if $\exists\, F_{[i]}, F_{[q]} \in Set_k$: $w_{[i]}/h_{[i]}(t) > w_{[q]}/h_{[q]}(t)$ then

calculate $t_1=\min\{t: w_{[i]}/h_{[i]}(t) = w_{[q]}/h_{[q]}(t)$ for every pair of jobs $F_{[i]}, F_{[q]} \in Set_k$

(*the shortest time required for 2 jobs different ratios to become equal*)

else t_1=A, A arbitrary large number

$t_2 = \min_{F_{[i]}\in Set_k} \left\{ C(\underline{s}_{[i]}(t)) \right\}$

(*This is fire's $F_{[i]}$'s shortest suppression completion time, i.e. $h_{[i]}(C(\underline{s}_{[i]}(t_2), t_2)) = 0$ *)

$t_{dec}=\min\{t_1, t_2, r_{k+1}-t\}$

(* Schedule $t_{dec}\sum_{j=1}^{K} s_{[i]j}(t)\rho_{[i]j}(t)$ "piece" of fire's $F_{[i]}$ suppression in the interval $[t, t+t_{dec}]$ according to McNaughton's wrap-around rule (change the assignment of resources to fires) *)

for i=1 to $|Set_k|$ do

if there is no resource "partially" empty and a non-busy resource exists then

assign fire $F_{[i]}$ to a non-busy resource

else

calculate for all processors the x unscheduled units of time until $t+t_{dec}$

if $P(\underline{s}_i(t+t_{dec}), t+t_{dec}) \le x$, then assign fire $F_{[i]}$ to the available processor with the highest containment rate

else

this resource is busy on $[t, t+t_{dec}-x]$

assign fire $F_{[i]}$ to the processor with the next highest containment rate for $[t, P(\underline{s}_i(t+t_{dec}), t+t_{dec})]$ (if possible) or $[t, t+t_{dec}-x]$ and then assign fire $F_{[i]}$ to the processor with the highest containment rate for $[t+t_{dec}-x, t+t_{dec}]$ (if not possible)

$t=t+t_{dec}$

for i=1 to $|Set_k|$

Calculate new $h_{[i]}(t)$

if $\exists\, F_{[i]} \in Set_k$: $h_{[i]}(t) > 0$ then

$Set_{k+1}=Set_{k+1} \cup \{F_{[i]}\}$

6. Conclusion

Scheduling the available fire fighting resources in the case of multiple forest fires may be treated as a job-scheduling problem with deteriorating job processing times and decreasing job values. The problem was stated as a multiprocessor jobs one. A model has been formulated with optimization criterion to maximize the total jobs' remaining value. A heuristic algorithm was presented in order to tackle this NP-hard problem.
The next research steps may include: a) implementation of the heuristic algorithm and testing its efficiency compared to real forest fires' data provided by Greek Fire Corps and b) use of $\Sigma w_i T_i$ or N_T as optimality criteria, where T_i is the tardiness (if any) in suppressing fire F_i and N_T the number of tardy fire containment efforts. Finally the model could be applied to other real life situations such as epidemics control.

References

[1] B. Alidaee (1993), Numerical Methods for Single machine Scheduling with Non-linear Cost Functions to Minimize Total Cost, *Journal of the Operational Research Society* **44**, 125 – 132.

[2] C. Buyukkoc, P.Varaiya, J. Walrand (1985), The cμ Rule Revisited, *Advanced Applied Probability* **17**, 237 – 238.

[3] Z. L. Chen (1996), Parallel machine scheduling with time dependent processing times, *Discrete Applied Mathematics* **70**, 81 – 93.

[4] Z-L. Chen (1997), Erratum to "parallel machine scheduling with time dependent processing times", *Discrete Applied Mathematics* **75**, 103.

[5] F. Della Croce, W. Szwarc, R. Tadei, P. Baracco, R. Di Tullio (1995), Minimizing the weighted sum of quadratic completion times on a single machine, *Naval Research Logistics* **42**, 1263 – 1270.

[6] M. Dimopoulou, I. Giannikos (2004), Towards an integrated framework for forest fire control, *European Journal of Operational Research* **152**(2), 476 – 486.

[7] M. Drozdowski (1996a), Real-time scheduling of linear speedup parallel machines, *Information processing Letters* **57**, 35 – 40.

[8] M. Drozdowski (1996b), Scheduling multiprocessor tasks- An overview, *European Journal of Operations Research* **94**, 215 – 230.

[9] M.L. Fisher and A.M. Krieger (1984), Analysis of a linearization Heuristic for Single-Machine Scheduling to Maximize Profit, *Mathematical Programming* **28**, 218 – 225.

[10] J. S. Fried, B. D. Fried (1996), Simulating wildfire containment with realistic tactics, *Forest Science* **42**(3), 267 – 281.

[11] Y. Hsieh and D.L. Bricker (1997), Scheduling linearly deteriorating jobs on multiple machines, *Computers and Industrial Engineering* **32**(4), 727 – 734.

[12] K. Inderfurth, G. Lindner, N.P. Rachaniotis (2005), Lotsizing in a Production System with Rework and Product Deterioration, *International Journal of Production Research* **43**(7), 1355 – 1374.

[13] T. Kawaguchi and S. Kyan (1986), Worst Case Bound of an LRF Schedule for the Mean Weighted Flow-Time Problem, *SIAM Journal of Computing* **15**, 1119 – 1129.

[14] J.K. Lenstra, R. Kan and P. Brucker (1977), Complexity of Machine Scheduling Problems, *Annals of Discrete Mathematics* **1**, 343 – 362.

[15] D. L. Martell, R. J. Drysdale, G. E. Doan and D. Boychuk (1984), An evaluation of forest fire initial attack resources, *Interfaces* **14**(5), 20 – 32.

[16] R. McNaughton (1959), Scheduling with deadlines and loss functions, *Management Science* **6**, 1 – 12.

[17] S.A. Mondal, A.K. Sen (2000), An improved precedence rule for single machine scheduling problems with quadratic penalty, *European Journal of Operational Research* **125**, 425 – 428.

[18] G. Mosheiov (1995), Scheduling jobs with step-deterioration: Minimizing makespan on a single and multi-machine, *Computing and Industrial Engineering* **28**, 869 – 879.

[19] G. Mosheiov (1998), Multi-machine scheduling with linear deterioration, *INFOR* **36**, 205 – 214.
[20] N.P. Rachaniotis (2004), Advances on job scheduling with time dependent parameters: theory and industrial applications, *Electronic proceedings of the 9th ELA doctorate workshop, 16th-18th June 2004, Monchy St. Eloi, France.*
[21] N.P. Rachaniotis, C.P. Pappis(2006), Scheduling Fire Fighting Tasks Using the Concept of "Deteriorating Jobs", *Canadian Journal of Forest Research* **36**, 652 – 658.
[22] G.D. Richards (1999), The Mathematical Modelling and Computer Simulation of Wildland Fire Perimeter Growth Over a 3-Dimensional Surface, *International Journal of Wildland Fire* **9**(3), 213 – 221.
[23] W.E. Smith (1956), Various Optimizers for Single-Stage Production, *Naval Research Logistics Quarterly* **3**, 59 – 66.
[24] R.H. Teunter and S.D.P. Flapper (2003), Lot-sizing for a single-stage single-product production system with rework of perishable production defectives, *OR Spectrum* **25**, 85 – 96.
[25] T.G. Voutsinas, C.P. Pappis (2002), Scheduling jobs with values exponentially deteriorating over time, *International Journal of Production Economics* **79** (3), 163 – 169.