J Sched (2008) 11: 457–473
DOI 10.1007/s10951-007-0033-9

Bicriteria robotic cell scheduling

Hakan Gultekin · M. Selim Akturk · Oya Ekin Karasan

Received: 1 May 2006 / Accepted: 11 June 2007 / Published online: 23 August 2007

Abstract This paper considers the scheduling problems arising in two- and three-machine manufacturing cells configured in a flowshop which repeatedly produces one type of product and where transportation of the parts between the machines is performed by a robot. The cycle time of the cell is affected by the robot move sequence as well as the processing times of the parts on the machines. For highly flexible CNC machines, the processing times can be changed by altering the machining conditions at the expense of increasing the manufacturing cost. As a result, we try to find the robot move sequence as well as the processing times of the parts on each machine that not only minimize the cycle time but, for the first time in robotic cell scheduling literature, also minimize the manufacturing cost. For each 1-unit cycle in two- and three-machine cells, we determine the efficient set of processing time vectors such that no other processing time vector gives both a smaller cycle time and a smaller cost value. We also compare these cycles with each other to determine the sufficient conditions under which each of the cycles dominates the rest. Finally, we show how different assumptions on cost structures affect the results.

Keywords Robotic cell · CNC · Bicriteria optimization · Controllable processing times

1 Introduction

The quest for improvement in component manufacturing has led to an increase in the level of automation in manufacturing industries. This trend involves the use of computer controlled machines and automated material handling devices. One of the widespread applications of automation is the use of robotic cells. A manufacturing cell consisting of a number of machines and a material handling robot is called a robotic cell. The efficient use of robotic cells necessitates the tackling of some important problems. Among these, the design of the cells and the scheduling of robot moves are prominent. In this paper we will consider the scheduling problems arising in two- and three-machine robotic cells producing identical parts. Each of the identical parts is assumed to have one distinct operation for each machine which makes a total of m operations in an m-machine robotic cell. These operations are assumed to be processed in the same sequence, namely in increasing order of machine label, for each part. In scheduling theory and practice, two main objectives are time and cost. Minimizing production time (equivalently, maximizing throughput) could have the highest priority in "production planning", while minimizing production costs has the highest priority in "process planning". It should also be noted that the former of these objectives is relevant when the demand is assumed to be unlimited. However, in today's highly competitive environment, most industries face a limited demand. Although there is an extensive literature on robotic cell scheduling problems, as far as the authors know, none of these consider cost objectives. Furthermore, the trade-offs involved in considering several different criteria provide useful insights to the decision maker. For example, a solution which minimizes the cycle time (long run average time to produce one part) may perform poorly in terms of cost. Thus, in the context of real life scheduling problems it is more relevant to consider problems with such dual criteria nature. This paper considers cost objectives simultaneously with time objectives in the context of robotic cells.

H. Gultekin · M.S. Akturk (✉) · O.E. Karasan
Department of Industrial Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey
e-mail: akturk@bilkent.edu.tr

In robotic cells, highly flexible Computer Numerical Control (CNC) machines are used for the metal cutting operations so that the machines and the robot can interact on a real time basis. Machining conditions such as the cutting speed and the feed rate are controllable variables for these machines. Consequently, the processing time of any operation on these machines can be reduced by changing the machining conditions at the expense of incurring extra cost resulting in the opportunity of reducing the cycle time. Due to this reasoning, assuming the processing times to be fixed on each machine is not realistic. In this study, the processing times are taken as decision variables. Different from the current literature, the problem is not only to find the robot move sequence but also to determine the processing times of the operations on the machines that simultaneously minimize the cycle time and the total manufacturing cost. Since we have two criteria, the optimal solution will not be unique but instead a set of nondominated solutions will be identified. A solution is called nondominated if no other feasible solution has smaller objective function values for both performance measures. Hoogeveen (2005) provides a review for multicriteria and bicriteria scheduling models.

The processing time for each operation can be optimized from two different points of view: (i) minimizing cost per unit, or (ii) maximizing production rate. The first criterion is common and basic to all manufacturing. On the other hand, in the current robotic cell scheduling literature only the second criterion (e.g., minimizing the overall cycle time or maximizing throughput) is discussed extensively. This objective is important when the production order must be completed as quickly as possible. When there is a limited demand, robotic cells should operate in the interval between these two cases (referred to as "high-efficiency range") that could be defined by generating a set of nondominated solutions by solving this bicriteria optimization problem.

A survey of the literature on controllable processing times covering the state of the art up through 1990 can be found in Nowicki and Zdrzalka (1995). Most of the studies in the existing literature of controllable processing times assume a linear cost function (Vickson 1980; van Wassenhove and Baker 1982; Daniels and Sarin 1989; Janiak and Kovalyov 1996; Cheng et al. 1998). Although this assumption simplifies the problem, it is not realistic because it does not reflect the law of diminishing returns. Thus, in this study we assume a nonlinear, strictly convex, differentiable cost function. Kayan and Akturk (2005) considered a single machine bicriteria scheduling model with controllable processing times. They selected total manufacturing cost and any regular scheduling measure-one which cannot be improved by increasing the processing times-such as makespan, completion time or cycle time, as the two objective criteria. They derived lower and upper bounds on processing times. In a related study, Shabtay and Kaspi (2004) considered the classical single machine scheduling problem of minimizing the total weighted flow time with controllable processing times. In their setting, the processing times can be controlled by allocating a continuously nonrenewable resource such as financial budget, overtime and energy. They assumed the processing times to be convex, nonlinear functions of the amount of the resource consumed. The objective was to allocate the resource to the jobs and to sequence the jobs so as to minimize the total weighted flow time.

There is an extensive literature on robotic cell scheduling problems with surveys including Crama et al. (2000) and Dawande et al. (2005). An n-unit cycle can be defined as a robot move cycle which produces exactly n units and ends up with the same state of the cell as the starting state. 1-unit cycles have drawn the bulk of the attention since they are practical, easy to understand and control. Furthermore, they are proved to give optimal solutions in two- (Sethi et al. 1992) and three-machine (Crama and Van de Klundert 1999) robotic cells producing identical parts. However, Brauner and Finke (1999) showed that 1-unit cycles need not be optimal for $m \geq 4$. Crama and Van de Klundert (1997) described a polynomial time dynamic programming algorithm for minimizing the cycle time over all 1-unit cycles in an m-machine cell producing identical parts. Akturk et al. (2005) considered a robotic cell with two identical CNC machines which possess operational and process flexibilities. Each part is assumed to have a number of operations, and the allocation of the operations to the machines along with the optimal robot move cycle that jointly minimize the cycle time were determined.

In this study we consider manufacturing cells used in metal cutting industries in which the CNC machines are used. As a consequence, the robots are used as material handling devices. However, in manufacturing cells used in chemical and electroplating industries, hoists are used as material handling devices. The production line consists of a sequence of chemical tanks. During the process, a part must be soaked on each chemical tank successively for a specified period of time. This problem is commonly known as hoist scheduling problem (see Chu 2006; Lee et al. 1997). The most distinct feature of the hoist scheduling problems from the robotic cell scheduling problems is that the job processing time at each machine is strictly limited by a lower and an upper bound (i.e., the time window constraints). This means that any hoist schedule that causes a hoist not to pick up a job within the time window is infeasible. In addition, after being removed from the tank, a part must be directly sent to the next tank on its route and submerged so that the time exposed to air is minimized. This is referred to as the no-wait constraints. However, in robotic cell scheduling problems, the parts can stay on the machines indefinitely after the processing is finished.

The organization of the paper is as follows: In the next section we will present the notation and some definitions that will be used throughout the paper. The problem definition and the mathematical formulation will also be presented in the next section. In Sect. 3 two- and three-machine cells will be analyzed and the set of nondominated solutions will be determined. In Sect. 4 different cost structures including the cost incurred by the robot will be analyzed. Section 5 is devoted to the concluding remarks and lists potential future research directions.

2 Problem definition

In this study we assume the cell to have an in-line robotic cell layout as can be seen in Fig. 1. There are no buffers in front of the machines. As a result, a part is either on one of the machines or on the robot at any time instant. As most of the studies of the robotic cell scheduling literature, we will restrict ourselves with 1-unit cycles since they are simple, practical and provably give good results. Each part is assumed to have a number of operations $o_1, o_2, \ldots, o_m$ in an m-machine robotic cell, where o_i represents the operation to be performed on machine i with corresponding processing time denoted by P_i. Processing times on the CNC machines can be written as functions of the machine parameters such as the cutting speed and the feed rate. As a consequence, selecting different parameters yields different processing time values. Total manufacturing cost for the CNC machines can be written as the summation of machining and tooling costs. The machining cost can be considered as a function of either the exact working time of the machines or the cycle time which includes some idle time for the machines. The former of these assumes that the machines incur cost only if they perform some operation on the parts. However, the latter case assumes that another job cannot be scheduled during these idle times. We will start with the former of these assumptions and the latter case will be analyzed in Sect. 4 where we consider different cost structures. There is a trade-off between machining and tooling costs in selecting the processing time values. Reducing the processing time reduces the machining cost but at the same time it reduces the tool life which in turn increases the tooling cost. Conversely, increasing the processing time increases the tool life and thus reduces the tooling cost, but this increases the machining cost.

Kayan and Akturk (2005) determined lower and upper bounds for the processing times in order to minimize a convex cost function and any regular scheduling measure. The lower bound of a processing time is derived from constraints such as the limited tool life, machine power and surface roughness. On the other hand, the upper bound of a processing time is the processing time value for which the total manufacturing cost is minimized, so that beyond this value of processing time, both objectives get worse. Note that, these upper and lower bounds are different from time window constraints used in hoist scheduling problems which indicate that any schedule that causes the hoist not to pick up a part within the time window is infeasible. In this study a schedule in which the processing times exceed their upper bounds is still feasible but proved to be not optimal. The lower bound corresponds to the minimum processing time-maximum cost case, whereas the upper bound corresponds to the maximum processing time-minimum cost case. Let P_i^L and P_i^U denote the lower and upper bounds for the processing time of operation o_i and $f_i(P_i)$ denote the manufacturing cost incurred by the same operation. In this study we assume $f_i(P_i)$ to be strictly convex and differentiable. As a consequence, from the derivation of the lower and upper bounds of the processing times, it is monotonically decreasing for $P_i^L \leq P_i \leq P_i^U$, $i = 1, 2, \ldots, m$. As a consequence, we can write the total manufacturing cost incurred by all the operations as $\sum_i f_i(P_i)$, which is also a convex,

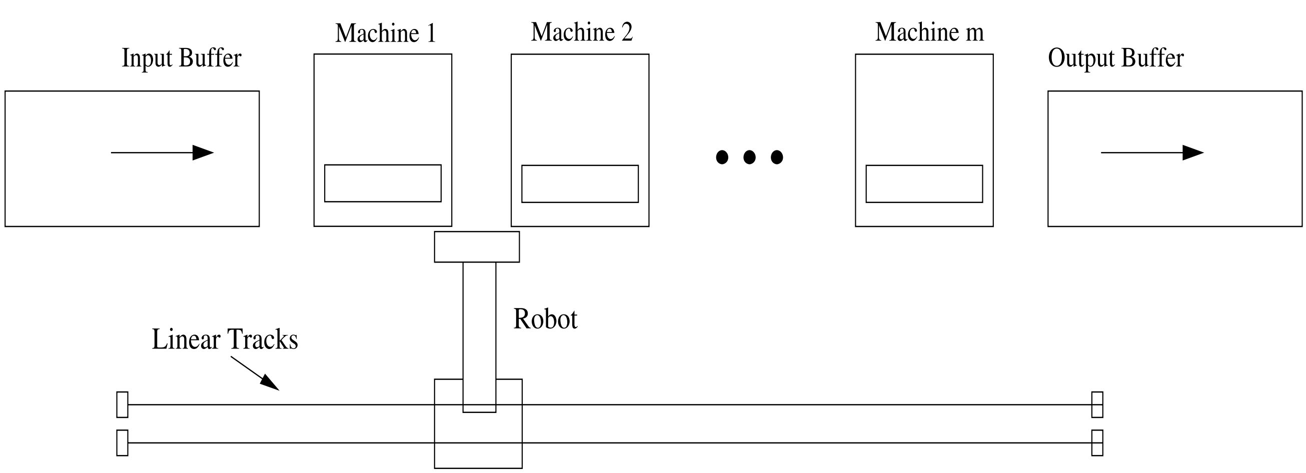

Fig. 1 Inline robotic cell layout

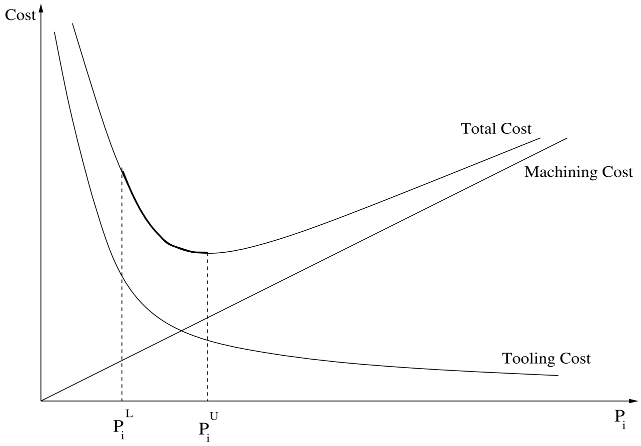

Fig. 2 Manufacturing cost with respect to processing time

differentiable function. Obviously, the total manufacturing cost does not depend on the robot move cycle but depends only on the processing times of the operations, whereas the cycle time depends on both. Figure 2 depicts the machining, tooling and the total manufacturing costs with respect to the processing time of an operation. P_i^L and P_i^U values and the cost function in between these values are also depicted. It is clear from the determination of the lower and the upper bounds that the portion of the manufacturing cost function lying in between the bounds is decreasing.

We denote a processing time vector as $\boldsymbol{P} = (P_1, P_2, \ldots, P_m)$. Any processing time violating one of its bounds is so called infeasible. As a consequence, we can define the set of feasible processing time vectors as $\mathcal{P}_{\text{feas}} = \{(P_1, P_2, \ldots, P_m) \in R^m : P_i^L \leq P_i \leq P_i^U, \ \forall i\}$. On the other hand, feasible robot move cycles are defined by Crama and Van de Klundert (1997) as the cycles in which the robot does not load an already loaded machine and does not unload an already empty machine. For example, in a two-machine robotic cell there are two feasible 1-unit cycles, namely, S_1^2 and S_2^2 where S_i^m represents the ith robot move cycle in an m-machine robotic cell. We denote the set of all feasible robot move cycles in an m-machine robotic cell as $\mathcal{S}_{\text{feas}}^m$. Before proceeding, let us present some definitions and notation that will be used throughout this study.

ϵ: The load/unload times of machines by the robot. Consistent with the literature we assume that loading/unloading times for all machines are the same.

δ: Time taken by the robot to travel between two consecutive machines. We assume that the robot travel time from machine i to j is $|j-i|\delta$. So the triangular equality is satisfied.

K: Cycle time, i.e., the long run average time that is required to produce one part.

C_o: Operating cost of the machines. Since we assume the machines to be identical, operating cost is the same for each machine.

T_i: Cost of tool i used (for $i = 1, \ldots, m$, since each operation might require a different tool).

$F_1(S, \boldsymbol{P}) = \sum_{i=1}^m f_i(P_i)$: Total manufacturing cost which depends only on the processing times. Note that the individual cost function $f_i(P_i)$ for each operation o_i, is strictly convex and differentiable, and it is monotonically decreasing for $P_i^L \leq P_i \leq P_i^U$, $i = 1, 2, \ldots, m$.

$F_2(S, \boldsymbol{P})$: Cycle time corresponding to robot move cycle S and processing time vector $\boldsymbol{P}$.

As a result of the bounding scheme explained above, we can formulate the bicriteria problem as follows:

min	*Total manufacturing cost*,
min	*Cycle time*,
Subject to	$P_i^L \leq P_i \leq P_i^U, \quad \forall i$

This formulation minimizes two conflicting objectives simultaneously. There are different ways to deal with bicriteria problems. We shall adopt the notation summarized in Hoogeveen (2005). Let f and g represent the two performance measures. The first method minimizes a linear composite objective function in f and g with unknown relative weights and is denoted by $G_l(f, g)$. The second way is called the *hierarchical optimization* or the *lexicographical optimization* and is denoted by $Lex(f, g)$. In this approach, performance measure f is assumed to be more important than g. As a result, this problem minimizes g subject to the constraint that the solution value of f is minimum. The third one is called the *epsilon-constraint method*, denoted by $\epsilon(f|g)$, as discussed in T'kindt and Billaut (2006). In this approach nondominated points are found by solving a series of problems of the form: minimize f given an upper bound on g. The epsilon-constraint method has been widely used in the literature, because the decision maker can interactively specify and modify the bounds and analyze the influence of these modifications on the final solution. The last approach (which will be used in this study) minimizes a composite objective function in f and g and is denoted by $G(f, g)$. In this approach all the nondominated points are generated where the only foreknowledge is that the composite function G is nondecreasing in both arguments. Since this particular approach does not use any of the aggregation methods, it is computationally more demanding than the other approaches.

The decision variables of the bicriteria problem formulated above are the processing times as well as the robot move cycles. In this study we will consider each 1-unit cycle individually. In other words, for each 1-unit cycle we will solve the bicriteria problem to determine the processing times and compare these 1-unit cycles with each other. However, in order to be able to find solutions minimizing both objectives simultaneously for 1-unit cycle S, we will first consider the epsilon-constraint formulation of the problem. That is, we will consider $\epsilon(F_1(S, \boldsymbol{P})|F_2(S, \boldsymbol{P}))$ to determine the sufficient conditions for the processing time values minimizing the manufacturing cost for a given level of cycle time. Using these conditions we will be able to write the manufacturing cost as a function of the cycle time, which means we will be able to determine the composite objective function G. As a result, for any given cycle time (manufacturing cost) value we will be able to determine the corresponding manufacturing cost (cycle time) value and the processing times of the parts on the machines.

Epsilon-Constraint Problem (ECP)

$$\begin{aligned} \min \quad & \textit{Total manufacturing cost}, \\ \text{Subject to} \quad & \text{Cycle time} \le K, && (1) \\ & P_i^L \le P_i \le P_i^U, \quad \forall i. && (2) \end{aligned}$$

In this study a solution to the bicriteria problem defines both a feasible robot move cycle and a corresponding feasible processing time vector for the parts. More formally, we can state a solution as follows:

Definition 1 A solution to the bicriteria problem for an m-machine robotic cell is represented as $\xi = (S^m, \boldsymbol{P})$ where $S^m \in \mathcal{S}^m_{\text{feas}}$ and $\boldsymbol{P} \in \mathcal{P}_{\text{feas}}$. Let $X = \{\xi = (S^m, \boldsymbol{P}) : S^m \in \mathcal{S}^m_{\text{feas}}$ and $P \in \mathcal{P}_{\text{feas}}\}$ be the set of all feasible solutions.

In the context of bicriteria optimization theory, solution ξ_1 dominates solution ξ_2 if it is not worse than ξ_2 under any of the performance measures, and is strictly better than it under at least one of the performance measures. Nondominated solutions are classified as Pareto optimal. We can state these more formally as follows:

Definition 2 We say that ξ_1 dominates ξ_2 and denote it as $\xi_1 \preceq \xi_2$ if and only if $F_1(\xi_1) \le F_1(\xi_2)$ and $F_2(\xi_1) \le F_2(\xi_2)$ one of which is a strict inequality. A solution $\xi^* \in X$ is called Pareto optimal, if there is no other $\xi \in X$ such that $\xi \preceq \xi^*$. If ξ^* is Pareto optimal, $z^* = (F_1(\xi^*), F_2(\xi^*))$ is called efficient. The set of all efficient points is the efficient frontier.

Recall that in this study the problem is twofold. That is, we try to find both the robot move sequence and the processing times of the parts on the machines. In order to achieve this, we will fix the robot move cycles and for each robot move cycle we will determine the set of nondominated processing time vectors. In other words, we will solve the bicriteria problem for each 1-unit cycle. The set of nondominated processing time vectors for an arbitrary 1-unit robot move cycle S_i^m can be defined as follows:

Definition 3 $P^*(S_i^m) = \{\boldsymbol{P} \in \mathcal{P}_{\text{feas}}$: There is no other $\overline{\boldsymbol{P}} \in \mathcal{P}_{\text{feas}}$ such that $(S_i^m, \overline{\boldsymbol{P}}) \preceq (S_i^m, \boldsymbol{P})\}$.

We already defined how one solution dominates another solution. However, while comparing robot move cycles with each other we will make use of the following, which defines how one robot move cycle dominates another one in the context of this study.

Definition 4 A cycle S_i^m is said to dominate another cycle S_j^m ($S_i^m \preceq S_j^m$) if there is no $\hat{P} \in P^*(S_j^m)$ such that $(S_j^m, \hat{P}) \preceq (S_i^m, \tilde{P}), \forall \tilde{P} \in P^*(S_i^m)$.

In the current literature, the processing times are assumed to be fixed. A cycle is said to dominate another one if the cycle time of the former is less than that of the latter with the same, fixed processing times used for both cycles. However, in order to find a dominance relation between two cycles as stated in Definition 4, the processing times used in the two cycles need not be the same. Hence, a dominance relation between two cycles is found by comparing the minimum cost values of the two cycles corresponding to the same cycle time value. That is, $F_1(S_i^m, \tilde{P})$ is compared with $F_1(S_j^m, \hat{P})$, for all $\tilde{P} \in P^*(S_i^m)$ and $\hat{P} \in P^*(S_j^m)$ where $F_2(S_i^m, \tilde{P}) = F_2(S_j^m, \hat{P})$. Although in such a flexible environment, 1-unit cycles may not be optimal, we will restrict ourselves with these cycles as is frequently done in the literature.

In the next section we will determine the set of nondominated processing time vectors for the 1-unit cycles for two- and three-machine cells.

3 Solution procedure

In this section we will consider two- and three-machine cells, respectively. For each 1-unit cycle, S, we will determine $P^*(S)$, the set of nondominated processing time vectors, and then compare these cycles with each other in light of Definition 4 to find sufficient conditions under which each of the cycles remain nondominated among all 1-unit cycles.

In order to define the robot move sequence performed under each cycle we will make use of the following definition which is borrowed from Crama and Van de Klundert (1997).

Definition 5 In an m-machine robotic cell, A_i is the robot activity defined as: robot unloads machine i, transfers part from machine i to machine $i+1$, loads machine $i+1$. The input buffer is denoted as machine 0 and the output buffer is denoted as machine $m+1$.

3.1 Two-machine case

Let us first analyze the S_1^2 cycle. The activity sequence of S_1^2 is $A_0A_1A_2$. The cycle time of this cycle can be calculated as $6\epsilon + 6\delta + P_1 + P_2$. In order to minimize the cost for a given cycle time value, K, the first constraint (1) of the ECP must be replaced by:

$$6\epsilon + 6\delta + P_1 + P_2 \leq K.$$

The following lemma is one of the major contributions of this study which determines $P^*(S_1^2)$, the processing times of the parts on each machine under the S_1^2 cycle that simultaneously minimize the cycle time and the total manufacturing cost. Let (P_1^*, P_2^*) be the optimal solution to the ECP formulated for the S_1^2 cycle, where the cycle time is bounded by K. Note that $(P_1^*, P_2^*) \in P^*(S_1^2)$, according to Definition 3.

Lemma 1

1. *If* $K = 6\epsilon + 6\delta + P_1^L + P_2^L$ *then* $P_1^* = P_1^L$ *and* $P_2^* = P_2^L$. *Corresponding cost is* $F_1(S_1^2, (P_1^L, P_2^L)) = f_1(P_1^L) + f_2(P_2^L)$.
2. *If* $K = 6\epsilon + 6\delta + P_1^U + P_2^U$ *then* $P_1^* = P_1^U$ *and* $P_2^* = P_2^U$. *Corresponding cost is* $F_1(S_1^2, (P_1^U, P_2^U)) = f_1(P_1^U) + f_2(P_2^U)$.
3. *If* $6\epsilon + 6\delta + P_1^L + P_2^L < K < 6\epsilon + 6\delta + P_1^U + P_2^U$ *then optimal processing times of the ECP are found by solving the following equations*: $6\epsilon + 6\delta + P_1^* + P_2^* = K$ *and* $\partial f_1(P_1^*) = \partial f_2(P_2^*)$.
   After solving, one may get one of the following cases:
   3.1 *If both processing times satisfy their own bounds then the solution found is optimal.*
   3.2 *Else if exactly one of the processing times,* P_i^*, *violates one of its bounds, say* P_i^b, *then the optimal solution is* $P_i^* = P_i^b$ *and* $P_j^* = K - 6\epsilon - 6\delta - P_i^b$, $i, j = 1, 2, i \neq j$.
   3.3 *Else if one of the processing times (assume* P_i^**) violates its lower bound (*P_i^L*) and the other one (*P_j^**) violates its upper bound (*P_j^U*) then the optimal solution is found by comparing the manufacturing costs of the following two processing time settings*:
      (i) $P_i^* = P_i^L$, $P_j^* = K - 6\epsilon - 6\delta - P_i^L$ *or*
      (ii) $P_j^* = P_j^U$, $P_i^* = K - 6\epsilon - 6\delta - P_j^U$, $i, j = 1, 2, i \neq j$.

Proof For S_1^2, the cycle time satisfies the following, $6\epsilon + 6\delta + P_1^L + P_2^L \leq K \leq 6\epsilon + 6\delta + P_1^U + P_2^U$. If $K = 6\epsilon + 6\delta + P_1^L + P_2^L$ then there exists a unique solution where $P_1^* = P_1^L$ and $P_2^* = P_2^L$, with corresponding cost $f_1(P_1^L) + f_2(P_2^L)$. In the same way, if $K = 6\epsilon + 6\delta + P_1^U + P_2^U$ then there exists a unique solution where $P_1^* = P_1^U$ and $P_2^* = P_2^U$, with corresponding cost $f_1(P_1^U) + f_2(P_2^U)$. For the remaining case, let (P_1^*, P_2^*) be the optimal solution to our problem. Then both of the following cannot hold at the same time: $P_i^* = P_i^L$ and $P_i^* = P_i^U$, unless $P_i^L = P_i^U$. Also since $6\epsilon + 6\delta + P_1^L + P_2^L < K < 6\epsilon + 6\delta + P_1^U + P_2^U$, either $P_1^* \neq P_1^L$ or $P_2^* \neq P_2^L$, and either $P_1^* \neq P_1^U$ or $P_2^* \neq P_2^U$. As a result, (P_1^*, P_2^*) is a regular point. Additionally, since the objective function and the constraints are convex, any point satisfying the Karush–Kuhn–Tucker (KKT) conditions is optimal. The Lagrangian function for point P^* is as follows:

$$L(P^*, \mu^*) = f_1(P_1^*) + f_2(P_2^*) + \mu^*(6\epsilon + 6\delta + P_1^* + P_2^* - K).$$

If we set $\nabla_P(L(P^*, \mu^*)) = 0$, we get: $\partial f_1(P_1^*) + \mu^* = 0$ and $\partial f_2(P_2^*) + \mu^* = 0$ with the additional constraints, $\mu^* \geq 0$ and $P_i^L \leq P_i^* \leq P_i^U$, $i = 1, 2$. As a result of these equations we have the following:

$$\mu^* = -\partial f_1(P_1^*) = -\partial f_2(P_2^*). \tag{3}$$

On the other hand, since $\partial f_i(P_i^*) < 0$ for $P_i^* < P_i^U \Rightarrow \mu^* = -\partial f_i(P_i^*) > 0$, which implies that the corresponding constraint must be satisfied as equality:

$$6\epsilon + 6\delta + P_1^* + P_2^* = K. \tag{4}$$

P_i^* can be found by solving (3) and (4) simultaneously. If exactly one of the P_i^* values violates one of its upper or lower bounds, P_i^* is set to the bound which is violated and the remaining processing time is found correspondingly using (4). Both of the processing times can also violate their own bounds. This can only be the case if one of the processing times violates its lower bound and the other one violates its upper bound. Let $P_i^* < P_i^L$ and $P_j^* > P_j^U$, $i, j = 1, 2$, $i \neq j$. Then there exist two alternative solutions, as stated in the statements (3.3.(i)) and (3.3.(ii)) of this lemma, and the optimal solution is found by comparing the manufacturing cost values for these two alternatives.

Note that, in order to determine the optimal processing time values, a nonlinear equation system must be solved (see (3) and (4)) which has a unique root for $P_i^* \geq 0, i = 1, 2$. The solution of these equations can be approximated by using either the Newtonian method, the golden search algorithm, or a bisection algorithm. □

The above solution procedure finds the processing times which give the minimum cost for a given cycle time value. That is, the given resource (in this case the cycle time) is allocated to two alternatives (in this case the processing times on the two machines) without violating the bounds. While allocating this resource, priority is given to the alternative (processing time) which has the highest contribution to the cost. That is, the processing time which has the highest contribution to the cost is increased more than the other one without exceeding the corresponding bounds. According to this lemma, for given manufacturing cost functions, $f_i(P_i)$, $\forall i$, the optimal processing times of the ECP can be written as functions of the cycle time, K. Using this fact, the total manufacturing cost can also be written as a function of K, which aids in determining the efficient frontier of the bicriteria problem. The range of the cycle time can easily be determined by using the lower and the upper bounds of the processing time values. As a result, the minimum manufacturing cost (cycle time) value corresponding to any given cycle time (manufacturing cost) value can be determined easily. Furthermore, the processing times of the parts on the machines can also be determined with the help of which the machine parameters such as the speed and the feed rate are determined.

Till now we considered the cost function to be any convex, nonlinear, differentiable function. Now let us consider more specifically a single-tool, single pass turning operation on CNC machines. For a more detailed explanation of the cost figures used in this part we refer the reader to Kayan and Akturk (2005). For this operation, the total manufacturing cost can be written as the summation of the machining and the tooling costs. Machining cost is $C_o \cdot (P_1 + P_2)$, where C_o is the operating cost of the CNC machine (\$/minute). Recall that in this section we assume the machining cost to be allocated in terms of the exact working times of the machines (P_1, P_2). Different allocation schemes will be analyzed in Sect. 4. On the other hand, the tooling cost is $T_1U_1P_1^{a_1} + T_2U_2P_2^{a_2}$, where $T_i > 0$ and $a_i < 0$ are specific constants for tool i and $U_i > 0$ is a specific constant for operation i regarding parameters such as the length and the diameter of the operation. We assume that each operation is performed with a corresponding tool. Let us consider a given cycle time value $K = 6\epsilon + 6\delta + P_1 + P_2 \Rightarrow P_1 + P_2 = K - 6\epsilon - 6\delta$. Then the machining cost can be rewritten as $C_o \cdot (K - 6\epsilon - 6\delta)$, which is constant for a given cycle time, K. In order to find the minimum total cost corresponding to K, the tooling cost will be minimized and summed with the corresponding machining cost. Then using Lemma 1 the solution can be found as follows:

1. If $K = 6\epsilon + 6\delta + P_1^L + P_2^L$ then $P_1^* = P_1^L$ and $P_2^* = P_2^L$. Corresponding cost is $C_o \cdot (K - 6\epsilon - 6\delta) + T_1U_1(P_1^L)^{a_1} + T_2U_2(P_2^L)^{a_2}$.
2. If $K = 6\epsilon + 6\delta + P_1^U + P_2^U$ then $P_1^* = P_1^U$ and $P_2^* = P_2^U$. Corresponding cost is $C_o \cdot (K - 6\epsilon - 6\delta) + T_1U_1(P_1^U)^{a_1} + T_2U_2(P_2^U)^{a_2}$.
3. Otherwise, P_i^* is found by solving the following two equations, $6\epsilon + 6\delta + P_1^* + P_2^* = K$ and $P_2^* = [(T_1U_1a_1)/(T_2U_2a_2)]^{1/(a_2-1)} \cdot (P_1^*)^{(a_1-1)/(a_2-1)}$. If any of the processing times violates any of the bounds, update all processing times accordingly so that they each satisfy their bounds and $6\epsilon + 6\delta + P_1^* + P_2^* = K$.

If the operations on both machines are made with a tool of the same type, then $a_1 = a_2 = a$ and $T_1 = T_2 = T$. In this case the above equations can be solved easily to determine the processing times as follows:

$$P_1^* = \big[(K - 6\epsilon - 6\delta)(U_2)^{1/(a-1)}\big] \Big/ \big[(U_1)^{1/(a-1)} + (U_2)^{1/(a-1)}\big]$$

and

$$P_2^* = \big[(K - 6\epsilon - 6\delta)(U_1)^{1/(a-1)}\big] \Big/ \big[(U_1)^{1/(a-1)} + (U_2)^{1/(a-1)}\big]$$

As a consequence, the optimal total cost can be written in terms of the cycle time as follows:

$$F_1 = C_o \cdot (K - 6\epsilon - 6\delta) + \Big\{\big[TU_1U_2(K - 6\epsilon - 6\delta)^a\big] \Big/ \big[(U_1)^{1/(a-1)} + (U_2)^{1/(a-1)}\big]^{a-1}\Big\}$$

for $6\epsilon + 6\delta + P_1^L + P_2^L \le K \le 6\epsilon + 6\delta + P_1^U + P_2^U$.

This identifies the whole set of nondominated solutions and shows the exact tradeoff between the cycle time K and the total manufacturing cost F_1.

Now let us consider the S_2^2 cycle for which the activity sequence can be written as $A_0A_2A_1$. The cycle time of S_2^2 can be calculated to be $\max\{6\epsilon + 8\delta, P_1 + 4\epsilon + 4\delta, P_2 + 4\epsilon + 4\delta\}$. Thus, constraint (1) of the ECP is replaced by the following:

$$\max\{6\epsilon + 8\delta, P_1 + 4\epsilon + 4\delta, P_2 + 4\epsilon + 4\delta\} \le K.$$

It is obvious that under S_2^2 K satisfies $\max\{6\epsilon + 8\delta, P_1^L + 4\epsilon + 4\delta, P_2^L + 4\epsilon + 4\delta\} \le K \le \max\{6\epsilon + 8\delta, P_1^U + 4\epsilon + 4\delta, P_2^U + 4\epsilon + 4\delta\}$. Restricting K to this region, the above constraint can be replaced by the following two linear constraints:

$$P_1 + 4\epsilon + 4\delta \le K,$$

$$P_2 + 4\epsilon + 4\delta \le K.$$

Lemma 2 *Under cycle S_2^2, for a given cycle time level K, the processing times minimizing the cost are: $P_i^* = \min\{P_i^U, K - 4\epsilon - 4\delta\}$, $i = 1, 2$.*

Proof Any point (P_1^*, P_2^*) for which $P_1^L < P_1^* < P_1^U$ and $P_2^L < P_2^* < P_2^U$ is a regular point and under these conditions $P_1^* = P_2^* = K - 4\epsilon - 4\delta$ is the point satisfying the KKT conditions. Since the objective function and the constraints are convex, this point is optimal. Including the bounds, the optimal processing times can be rewritten as $P_i^* = \min\{P_i^U, K - 4\epsilon - 4\delta\}$, $i = 1, 2$. As one can observe, for any nonlinear, convex cost function we get the same processing time values. □

As a consequence of this lemma, the total manufacturing cost can be written as a function of the cycle time which defines the efficient frontier of the bicriteria problem. The intuition behind this lemma is the following: Having greater processing times without exceeding the upper bounds of the processing times and the given cycle time level K is better in terms of manufacturing cost and in the above case the processing times are set to their maximum allowable level. Note that in this cycle, after loading a part to one of the machines the robot does not wait in front of the machine but instead performs other activities and returns back to unload the part after finishing these activities. Then, if the processing of a part finishes before the robot returns back to unload the part, the speed of the machine can be reduced so that the processing time is increased without increasing the cycle time. This means having less cost with the same cycle time value. Furthermore, it is apparent that the optimal processing times on both machines are balanced under the S_2^2 cycle. A numerical example will be helpful for understanding.

Example 1 Let us consider S_2^2 cycle and a turning operation for this example and assume that both machines use a tool of the same type. Let the parameters be given as follows: $T = 4$, $C_o = 0.5$, $U_1 = 0.2$, $U_2 = 0.03$, $a = -1.43423$, $P_1^L = 0.5$, $P_1^U = 1.4$, $P_2^L = 0.3$, $P_2^U = 0.64$, $\epsilon = 0.1$, and $\delta = 0.2$. Let us first consider the solution where all of the processing times are set to their lower bounds, $(S_2^2, (0.5, 0.3))$. The Gantt chart on top of Fig. 3 depicts this cycle. For this solution, $F_1(S_2^2, (0.5, 0.3)) = 3.237$ and $F_2(S_2^2, (0.5, 0.3)) = 2.2$. If we analyze this cycle, we observe that the robot never waits and is the bottleneck for this case. Without increasing the cycle time, we can increase the processing time on the first machine from 0.5 to 1 and the processing time on the second machine from 0.3 to 1. Now let us find the optimal processing times on these machines for $K = 2.2$ by using Lemma 2. We have $P_i^* = \min\{P_i^U, K - 4\epsilon - 4\delta\} \Rightarrow P_1^* = 1, P_2^* = 0.64$. That is, the processing time of the first machine is increased up to the end of the idle time period shown in Fig. 3, but the processing time of the second machine could not be increased because the upper bound of this processing time is less than this value. The Gantt chart for this solution is depicted as the second chart in Fig. 3. As it is seen, for this case the robot never waits and $F_1(S_2^2, (1, 0.64)) = 1.848$ and $F_2(S_2^2, (1, 0.64)) = 2.2$. From Definition 4, we conclude that $(S_2^2, (1, 0.64)) \preceq (S_2^2, (0.5, 0.3))$. Thus, we eliminate $(S_2^2, (0.5, 0.3))$ from further consideration. Let us also consider another solution in which all of the processing times are fixed to their upper bounds, $(S_2^2, (1.4, 0.64))$. The Gantt chart for this solution is depicted as the last one in Fig. 3. Note that in this case the first machine becomes the bottleneck and the robot waits for this machine in order to finish the processing. In this case, $F_1(S_2^2, (1.4, 0.64)) = 1.7413$ and $F_2(S_2^2, (1.4, 0.64)) = 2.6$. When we compare this solution with $(S_2^2, (1, 0.64))$, $F_1(S_2^2, (1.4, 0.64)) < F_1(S_2^2, (1, 0.64))$ but $F_2(S_2^2, (1.4, 0.64)) > F_2(S_2^2, (1, 0.64))$. That is, none of these two solutions dominates one another.

After characterizing $P^*(S_1^2)$ and $P^*(S_2^2)$, the following theorem compares the two 1-unit robot move cycles S_1^2 and S_2^2 and finds the sufficient conditions under which one dominates the other.

Theorem 1 *Whenever S_2^2 is feasible ($K \geq 6\epsilon + 8\delta$) it dominates S_1^2.*

Proof The cycle time of the S_2^2 cycle can be at least $6\epsilon + 8\delta$. Hence, for the cycle time values less than $6\epsilon + 8\delta$, S_2^2 cycle is not feasible and we have $S_1^2 \preceq S_2^2$. Now let us consider the region where the cycle time is at least $6\epsilon + 8\delta$ and compare the two cycles for the same cycle time value. Let $(\hat{P}_1, \hat{P}_2) \in P^*(S_1^2)$, which satisfies $K = \hat{P}_1 + \hat{P}_2 + 6\epsilon + 6\delta$. The optimal processing times for S_2^2 with the same cycle time value are the following: $\tilde{P}_i = \min\{P_i^U, \hat{P}_1 + \hat{P}_2 + 2\epsilon + 2\delta\}$, $i = 1, 2$, where $(\tilde{P}_1, \tilde{P}_2) \in P^*(S_2^2)$. Since $P_i^U \geq \hat{P}_i$ and $\hat{P}_1 + \hat{P}_2 + 2\epsilon + 2\delta \geq \hat{P}_i$, then $\tilde{P}_i \geq \hat{P}_i$. For $P_i^L \leq P_i^* \leq P_i^U$, the total manufacturing cost is monotonically decreasing. Since for the same cycle time value, the optimal processing times for the S_2^2 cycle are greater than that of the S_1^2 cycle, that means the total manufacturing cost of the S_2^2 cycle is less than that of S_1^2 cycle. Consequently, we have $S_2^2 \preceq S_1^2$. □

This theorem is one of the major contributions of this paper and states that for a given cell data, for the cycle time values that can be attained by the S_2^2 cycle, the minimum cost is also attained by the same cycle. However, for very small cycle time values which cannot be attained by the S_2^2 cycle, although the cost values can be very high, S_1^2 cycle is still an alternative for the decision maker. Note that $K < 6\epsilon + 8\delta \Leftrightarrow \hat{P}_1 + \hat{P}_2 < 2\delta$. This fact can be used to rewrite the above theorem. According to the different values of the bounds of the processing times, different versions of this theorem can also be created. For example, if $P_1^L + P_2^L \geq 2\delta$ then all the cycle time values that can be attained by the S_1^2 cycle can also be attained by the S_2^2 cycle. As a result, $S_2^2 \preceq S_1^2$ in the whole region. In a similar way if $P_1^U + P_2^U < 2\delta$ then $S_1^2 \preceq S_2^2$ in the whole region. From these, we can conclude that for greater processing times S_2^2 is preferable to S_1^2 and for smaller processing times vice versa. Observe that $K = 6\epsilon + 8\delta$ or $\hat{P}_1 + \hat{P}_2 = 2\delta$, is the region of indifference in the case of Sethi et al. (1992). However, in the settings of this study, in this region S_2^2 dominates S_1^2. That is, previous studies can not handle the cost

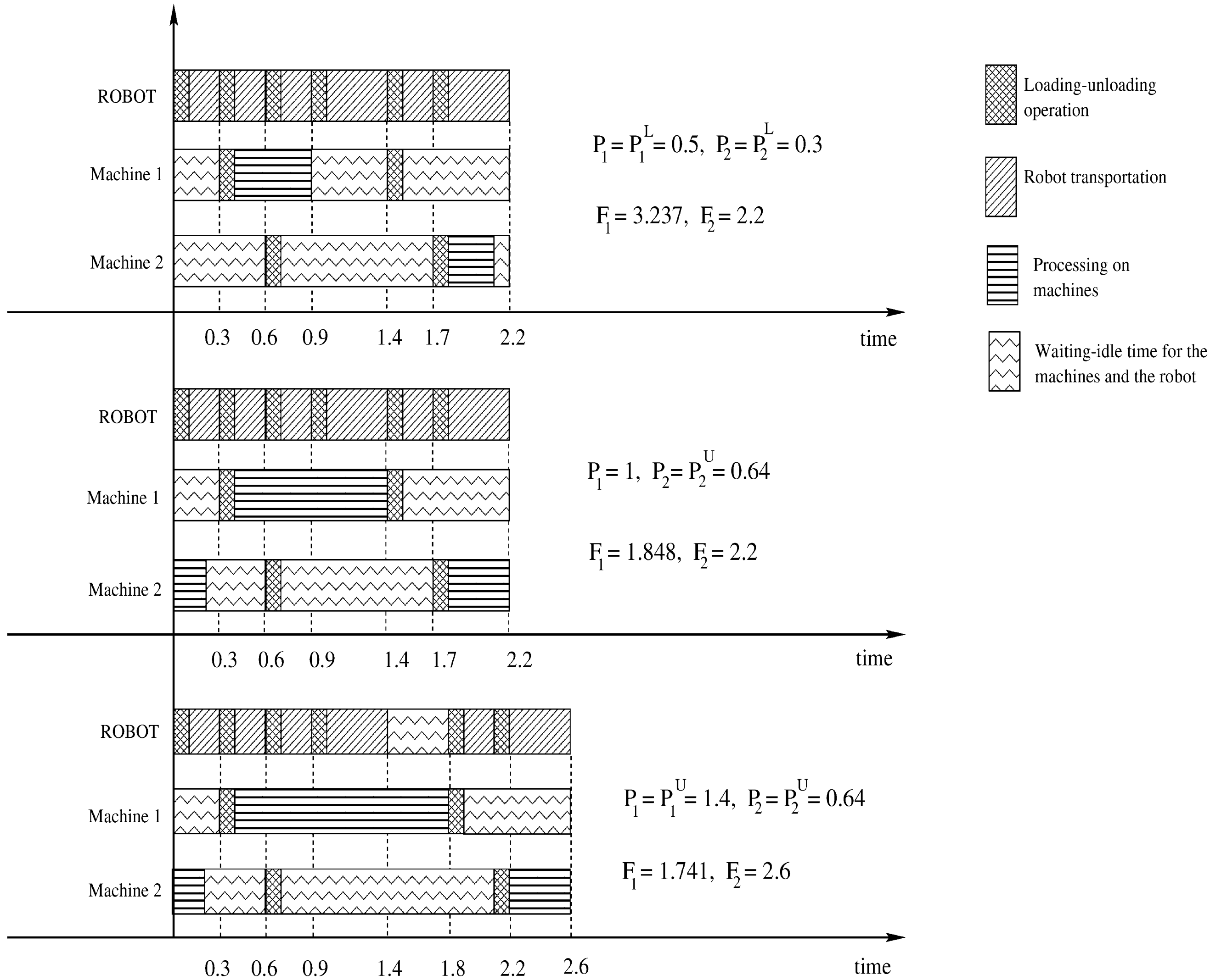

Fig. 3 Gantt charts for different processing time settings for the above example

component and thus state that both cycles perform identically. However, although both cycles give the same cycle time value, S_2^2 has a smaller manufacturing cost value and is preferred to S_1^2. Additionally, if $P_1^U \leq 2\epsilon + 4\delta$ and $P_2^U \leq 2\epsilon + 4\delta$ then the cycle time of S_2^2 can only take one value which is equivalent to $6\epsilon + 8\delta$.

The following example will aid in depicting such special cases.

Example 2 Let us consider a turning operation and assume that both machines use a tool of the same type. Let the parameters be given as follows: $T = 4$, $C_o = 0.5$, $U_1 = 0.2$, $U_2 = 0.03$, $a = -1.43423$, $P_1^L = 0.1$, $P_1^U = 1.4$, $P_2^L = 0.08$, $P_2^U = 0.64$, and $\epsilon = 0.02$. In order to present different occurrences of the efficient frontier, four different values are used for δ. Using Lemmas 1 and 2, the efficient frontiers for these two cycles are drawn in Fig. 4. In the first case, let $\delta = 0.1$. As a result, $P_1^L + P_2^L < 2\delta < P_1^U + P_2^U$. The bold curves show that for $K < 6\epsilon + 8\delta = 0.92$, $S_1^2 \preceq S_2^2$ and, otherwise, $S_2^2 \preceq S_1^2$. Although the cost of the S_1^2 cycle for $K < 6\epsilon + 8\delta$ is very high, the cycle time is smaller than that of S_2^2 and this region is still an alternative for the decision maker. In the second case, let $\delta = 0.08$, which results in $P_1^L + P_2^L \geq 2\delta$. As it is seen from the figure, for all cycle time and cost combinations, S_2^2 is preferable to S_1^2. In the third case, let $\delta = 1.1$. In this case, $P_1^U + P_2^U \leq 2\delta$ and the only cycle time value that S_2^2 can take is equivalent to $6\epsilon + 8\delta = 8.92$. The minimum cost corresponding to this value of cycle time is found by setting $P_i^* = P_i^U$, $i = 1, 2$. On the other hand, when the same processing time settings are used for the S_1^2 cycle, the cycle time becomes 8.76. Since the same processing time values are used, the cost is the same for both cycles. Thus, we conclude that in this case $S_1^2 \preceq S_2^2$. Lastly, let $\delta = 0.4$. Since $P_1^U \leq 2\epsilon + 4\delta$ and $P_2^U \leq 2\epsilon + 4\delta$, the cycle time of S_2^2 can only be $6\epsilon + 8\delta = 3.32$ and this cycle time value cor-

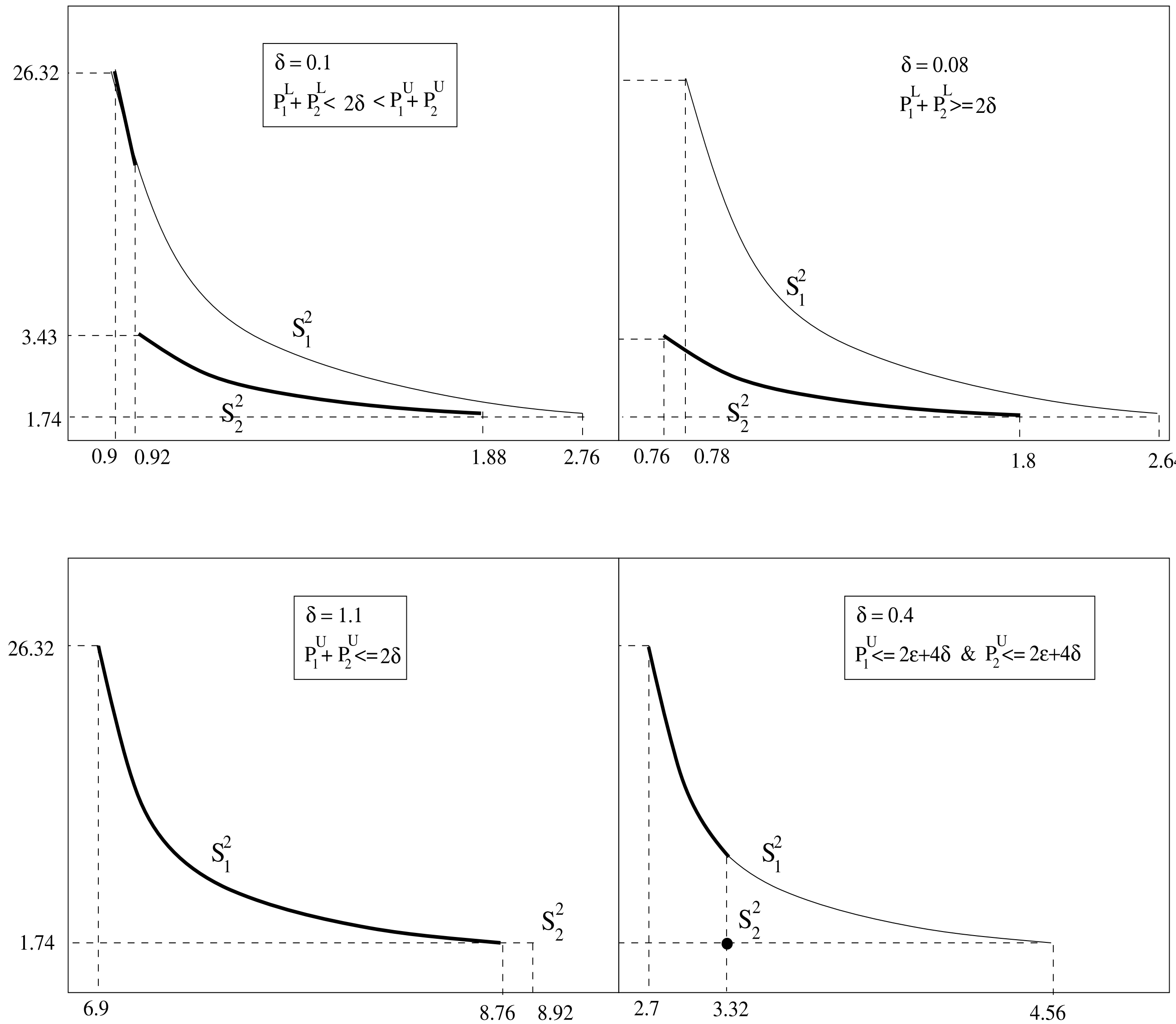

Fig. 4 Different occurrences of the efficient frontier with respect to given parameters

responds to a cost of 1.74. S_1^2 cannot take a cost value less than 1.74. As a result, S_2^2 dominates S_1^2 unless the cycle time of $S_1^2 < 3.32$. For cycle time values smaller than 3.32, the only alternative is the S_1^2 cycle.

Akturk et al. (2005) proved that 1-unit cycles need not be optimal in the whole region even in two-machine robotic cells producing identical parts with single objective function when the processing times on the machines are not assumed to be fixed. They assumed that each part has a set of operations and both machines are capable of performing all the operations. Then the processing time of a part on a machine depends on the operations allocated to that machine, and is assumed to be a decision variable. The following example provides a similar result for this study by finding a processing time setting for a 2-unit cycle, in which for the same cycle time value the 2-unit cycle gives the minimum cost. Note that in a two-machine cell just after loading a part to the second machine the robot can either wait in front of the machine to finish processing of the part or can return back to the input buffer to take another part. This is the only state where a transition from S_1^2 to S_2^2 or from S_2^2 to S_1^2 can happen. The transition moves of the robot from S_1^2 (resp., S_2^2) to S_2^2 (resp., S_1^2) is denoted as S_{12} (resp., S_{21}) (Hall et al. 1997). Under S_{12} (resp., S_{21}), the robot uses cycle S_1^2 (resp., S_2^2) during processing of part i on the first machine, and cycle S_2^2 (resp., S_1^2) during processing on the second machine. The only 2-unit cycle in a two-machine cell is denoted as $S_{12}S_{21}$ and has the following activity sequence: $A_0A_1A_0A_2A_1A_2$. One repetition of this cycle produces two parts. We will determine the processing times of these two parts on the machines which can be different from each other. In order to denote this, let P_{ij} represent the processing time on machine i for part j where $i = 1, 2$ and $j = 1, 2$. The cycle time for this cycle can be derived to be:

$$\frac{1}{2}\max\{P_{11} + P_{22} + 12\epsilon + 14\delta,\\ P_{11} + P_{22} + P_{12} + 10\epsilon + 10\delta,\\ P_{11} + P_{22} + P_{21} + 10\epsilon + 10\delta\}.$$

Example 3 Let us consider the turning operation again and use the same parameter values for this example as the one above with $\delta = 0.1$. Let $P_{11} = 0.1$, $P_{12} = 0.44$, $P_{21} = 0.44$, and $P_{22} = 0.08$. With these settings, the cycle time of $S_{12}S_{21}$ is 0.91 and the total manufacturing cost is 6.325. Since the cycle time of S_2^2 cannot take values less than 0.92 with these parameters, $S_{12}S_{21} \preceq S_2^2$. On the other hand, for $K = 0.91$, the minimum cost for S_1^2 is $9.214 \Rightarrow S_{12}S_{21} \preceq S_1^2$.

This example shows that 1-unit cycles need not be optimal, even for two-machines, under the assumptions of this study. The following theorem determines the regions where they are optimal. Within the proof of this theorem, we also provide a lower bound on the cycle time for the regions where the dominance of the specified two cycles cannot be attained.

Theorem 2 *S_2^2 dominates all other robot move cycles whenever it is feasible ($K \geq 6\epsilon + 8\delta$) and S_1^2 dominates all other robot move cycles for $K < 6\epsilon + 7\delta$.*

Proof In any feasible robot move cycle, in order to produce one part the robot must at least perform the following set of activities: The robot loads and unloads both machines exactly once, (4ϵ), also takes a part from the input buffer, (ϵ), and drops a part to the output buffer, (ϵ). Then the total load and unload time is exactly 6ϵ. As the forward movement, the robot travels all the way from the input buffer to the output buffer in some sequence of robot activities, which takes at least 3δ and in order to return back to the initial state the robot must travel back to the input buffer which again takes at least 3δ. Thus, the travel time is at least 6δ. On the other hand, after loading a part to machine i, the robot has two options: It either waits in front of the machine, (P_i), or travels to another machine to make some other activities which takes at least δ time units. Then, for both machines, we have $\min\{P_1, \delta\} + \min\{P_2, \delta\}$. Furthermore, since $\min\{P_1 + P_2, \delta\} \leq \min\{P_1, \delta\} + \min\{P_2, \delta\}$, we have the following:

$$6\epsilon + 6\delta + \min\{P_1 + P_2, \delta\}. \tag{5}$$

On the other hand, the cycle time of any cycle is greater than the time between two consecutive loadings of a machine for which the consecutive loading time is the greatest. But in order to make a consecutive loading, the robot must at least perform the following activities: After loading a part to some machine i, the robot either waits in front of the machine or makes some other activities which takes at least P_i amount of time; then, the robot unloads machine i, (ϵ); transports the part to machine $(i + 1)$, (δ); loads it, (ϵ); returns back to machine $(i - 1)$, (2δ); unloads it, (ϵ); transports the part to machine i, (δ); and loads it, (ϵ). This in total makes $4\epsilon + 4\delta + P_i$. In order to find the greatest consecutive loading time we take $\max\{P_1, P_2\}$. As a result we have the following:

$$4\epsilon + 4\delta + \max\{P_1, P_2\}. \tag{6}$$

Combining (5) and (6) we can conclude that in order to produce one part with any robot move cycle, the robot requires the following amount of time at the least:

$$\max\big\{6\epsilon + 6\delta + \min\{P_1 + P_2, \delta\}, 4\epsilon + 4\delta + \max\{P_1, P_2\}\big\}. \tag{7}$$

Observing this equation we can state that for any given cycle time K, if $K < 6\epsilon + 7\delta$ then $P_1 + P_2 < \delta$. As a consequence, $K = 6\epsilon + 6\delta + P_1 + P_2$ which is equivalent to the cycle time of the S_1^2 cycle. This concludes that for any given cycle time $K < 6\epsilon + 7\delta$, S_1^2 cycle dominates all other cycles. On the other hand, if $K \geq 6\epsilon + 7\delta$, the processing times can at most be increased (the cost can be reduced) so that $4\epsilon + 4\delta + \max\{P_1, P_2\} \leq K$. From here, since the processing times have upper bounds, the processing times satisfying the minimum cost for a given cycle time K are as follows, $P_i = \max\{P_i^U, K - 4\epsilon - 4\delta\}$, $i = 1, 2$. Using this processing time setting, we get a lower bound for the cost for given cycle time value K. From Lemma 2, this is equivalent to the processing time setting that minimizes the cost for given K under the S_2^2 cycle. However, S_2^2 cycle is feasible for $K \geq 6\epsilon + 8\delta$. This completes the proof. □

This theorem determines the regions of optimality for the two 1-unit cycles. It is also shown that for $6\epsilon + 7\delta \leq K < 6\epsilon + 8\delta$, 1-unit cycles need not be optimal. Note that in this region S_2^2 is not feasible. In order to determine the worst case performance of the S_1^2 cycle inside this region one can calculate the processing times for the S_1^2 cycle according to Lemma 1 and compare the cost corresponding to this setting of processing times with the cost corresponding to setting $P_i = \max\{P_i^U, K - 4\epsilon - 4\delta\}$, $i = 1, 2$, to get a lower bound for the cost.

The next section is devoted to the three-machine robotic cells.

3.2 Three-machine case

Increasing the number of machines in a robotic cell increases the number of feasible robot move cycles, drastically. More specifically, Sethi et al. (1992) proved that the number of feasible 1-unit cycles for an m-machine robotic cell is $m!$. For a three-machine robotic cell there are a total of six feasible 1-unit cycles. The robot activity sequences and the corresponding cycle times for these cycles are presented in the Appendix. Note that S_1^3 cycle is very similar, with respect to robot activity sequence and the cycle time formula, to S_1^2 cycle which may both be classified as the forward cycles and the S_6^3 cycle is very similar, again for similar properties, to S_2^2 cycle both of which may be classified as the backward cycles. Let us first consider the forward move cycle, S_1^3. Proceeding just as with S_1^2, we can solve the single criterion problem. But this time we have three variables to determine, P_1, P_2, and P_3. The following lemma determines the optimal processing times for the ECP of the S_1^3 cycle.

Lemma 3 *Let* (P_1^*, P_2^*, P_3^*) *be the optimal processing times for the ECP formulated for the* S_1^3 *cycle for a given cycle time,* K. *Then this point satisfies the following set of nonlinear equations*:

$$\partial f_1(P_1^*) = \partial f_2(P_2^*) = \partial f_3(P_3^*),$$

$$P_1^* + P_2^* + P_3^* = K - 8\epsilon - 8\delta.$$

After solving,

1. *If all of the processing times satisfy their lower and upper bounds, then the solution found is optimal.*
2. *Else, if for exactly one index* i, $i = 1, 2, 3$, P_i^* *violates its bounds, set it to the bound which is violated. Let* P_i^b *represent the bound which is violated. Update* K *such that* $\hat{K} = K - P_i^b$. *In order to determine the remaining two processing times, proceed just as solving the* S_1^2 *cycle case with cycle time set to* $\hat{K}$.
3. *Else, if exactly two processing times violate their bounds and if both violate their lower or both violate their upper bounds then set them to their own violated bound. That is, for* $i, j, k = 1, 2, 3$, $i \neq j$, $i \neq k$, $j \neq k$, *if* $P_i^* < P_i^L$ *and* $P_j^* < P_j^L$ *(or* $P_i^* > P_i^U$ *and* $P_j^* > P_j^U$*), set* $P_i^* = P_i^L$ *and* $P_j^* = P_j^L$ *(*$P_i^* = P_i^U$ *and* $P_j^* = P_j^U$*). The last processing time is found as* $P_k^* = K - P_i^L - P_j^L - 8\epsilon - 8\delta$ *(*$P_k^* = K - P_i^U - P_j^U - 8\epsilon - 8\delta$*). Else, if one of the processing times violates its lower (assume w.l.o.g* $P_i^* < P_i^L$*) and the other one violates its upper bound (w.l.o.g* $P_j^* > P_j^U$, $i \neq j$*) then compare the manufacturing costs found by the following two alternatives*:
   (i) *Set* $P_i^* = P_i^L$ *and solve for the remaining two processing times similar to the* S_1^2 *cycle,*
   (ii) *Set* $P_j^* = P_j^U$ *and solve for the remaining two processing times similar to the* S_1^2 *cycle.*
4. *Else, if all processing times violate their own bounds, let* P_i^b *represent the violated bound for* P_i^*, $i = 1, 2, 3$. *Compare the manufacturing costs for the following three alternative solutions*:
   (i) *Set* $P_1^* = P_1^b$ *and solve for the remaining two processing times similar to* S_1^2 *case,*
   (ii) *Set* $P_2^* = P_2^b$ *and solve for the remaining two processing times similar to* S_1^2 *case,*
   (iii) *Set* $P_3^* = P_3^b$ *and solve for the remaining two processing times similar to* S_1^2 *case.*

Proof The proof is very similar to that for cycle S_1^2 and is omitted here. □

Now let us consider the backward cycle, S_6^3. For this case, the cycle time K can take values between $\max\{8\epsilon + 12\delta, P_1^L + 4\epsilon + 4\delta, P_2^L + 4\epsilon + 4\delta, P_3^L + 4\epsilon + 4\delta\} < K < \max\{8\epsilon + 12\delta, P_1^U + 4\epsilon + 4\delta, P_2^U + 4\epsilon + 4\delta, P_3^U + 4\epsilon + 4\delta\}$.

Lemma 4 *Under cycle* S_6^3, *the optimal processing times for the ECP are* $P_i^* = \min\{P_i^U, K - 4\epsilon - 4\delta\}$, $i = 1, 2, 3$.

Proof The proof is very similar to the S_2^2 case and is omitted here. □

In the following two theorems, we will prove some dominance relations of the type stated in Definition 4. Considering only the 1-unit cycles, Crama and Van de Klundert (1997) proved that the set of pyramidal permutations necessarily contains an optimal solution of the problem. Let $A_0, A_{i_1}, \ldots, A_{i_k}, A_{i_{k+1}}, \ldots, A_{i_m}$ denote the activity sequence of a 1-unit cycle in an m-machine cell. Then Crama and Van de Klundert (1997) defines this cycle to be pyramidal if $1 \le i_1 < \cdots < i_k = m$ and $m > i_{k+1} > \cdots > i_m \ge 1$. The following is an important theorem which proves that the classical dominance of pyramidal permutations is valid with the assumptions of this study as well.

Theorem 3 *The set of pyramidal cycles is dominating among* 1*-unit cycles.*

Proof According to Theorem 3 of Crama and Van de Klundert (1997), for any processing time setting there exists at least one pyramidal cycle which gives a smaller cycle time than any nonpyramidal cycle. This means that, for any processing time setting there exists at least one pyramidal cycle which has the same cost value with the nonpyramidal cycle but with a smaller cycle time value meaning that the nonpyramidal cycle is dominated. Note that this processing time setting need not be optimal for the pyramidal cycle for this cost value. That is, with another processing time setting for the pyramidal cycle a smaller cycle time value can be found which corresponds to the same cost value. This completes the proof. □

In three-machine cells, the S_2^3 and S_4^3 cycles are nonpyramidal and the remaining ones are pyramidal. According to the theorem above these two cycles are dominated and can be eliminated from further consideration. In the following theorem we will compare the remaining cycles and determine the regions where S_6^3 dominates the remaining cycles. In order to prove these, we will select an arbitrary K and find the optimal processing times for the ECP formulation for each cycle and compare them with each other. Let $P_i^*(S_j^3)$ denote the nondominated processing times on machine i for the robot move cycle S_j^3.

Theorem 4 *Whenever S_6^3 is feasible ($K \geq 8\epsilon + 12\delta$), it dominates all of the remaining cycles.*

Proof S_6^3 cannot take cycle time values less than $8\epsilon + 12\delta$. Thus, for an arbitrary selection of $K \geq 8\epsilon + 12\delta$, we will compare S_6^3 with S_1^3, S_3^3 and S_5^3 in the following cases respectively:

1. The optimal solution of ECP for S_1^3 satisfies $K = 8\epsilon + 8\delta + P_1^*(S_1^3) + P_2^*(S_1^3) + P_3^*(S_1^3)$. Optimal processing time values of ECP for the S_6^3 corresponding to this cycle time value can be found by using Lemma 4 as $P_i^*(S_6^3) = \min\{P_i^U, 4\epsilon + 4\delta + P_1^*(S_1^3) + P_2^*(S_1^3) + P_3^*(S_1^3)\} \geq P_i^*(S_1^3)$. Therefore, $S_6^3 \preceq S_1^3$ in this region.
2. The optimal solution of ECP for S_3^3 satisfies $K = 4\epsilon + 4\delta + \max\{P_3^*(S_3^3), P_1^*(S_3^3) + 4\epsilon + 6\delta, P_1^*(S_3^3) + P_2^*(S_3^3) + 2\epsilon + 2\delta\}$. Optimal processing time values of ECP for the S_6^3 corresponding to this cycle time value can be found by using Lemma 4 as $P_i^*(S_6^3) = \min\{P_i^U, K - 4\epsilon - 4\delta\} = \min\{P_i^U, \max\{P_3^*(S_3^3), P_1^*(S_3^3) + 4\epsilon + 6\delta, P_1^*(S_3^3) + P_2^*(S_3^3) + 2\epsilon + 2\delta\}\} \geq P_i^*(S_3^3)$. Thus, $S_6^3 \preceq S_3^3$ for $K \geq 8\epsilon + 12\delta$.
3. As one can observe from the cycle time functions presented in the Appendix the cycle time function of S_5^3 is very similar to that of S_3^3. If we swap the places of $P_1^*(S_3^3)$ and $P_3^*(S_3^3)$ we get one another. Therefore, this case is identical with case 2, the only difference being the places of $P_1^*(S_3^3)$ and $P_3^*(S_3^3)$. □

This theorem derives a similar result to Theorem 1 for three-machine cells. According to this theorem, for a given cell data such as the loading/unloading time and robot transportation time, the backward cycle gives the minimum cost values for the cycle time values that can be attained by this cycle. The remaining three cycles, S_1^3, S_3^3, and S_5^3 can only be optimal for the cycle time values that cannot be attained by S_6^3. Sethi et al. (1992) provided a decision tree on conditions for the robot move cycles to be optimal with any given cell data considering only the cycle time. However, the above theorem shows that earlier results are not valid anymore when the manufacturing cost is considered besides the cycle time. That is, considering the cycle time as the only objective hinders the additional insights provided by the cost of the suggested settings for the cell.

Let us now consider the region for $K < 8\epsilon + 12\delta$. In this region three cycles remain nondominated. According to the cycle times of these cycles presented in the Appendix, one can easily verify that under cycle S_1^3, $K \geq 8\epsilon + 8\delta + P_1^L + P_2^L + P_3^L$. Similarly, for cycle S_3^3, $K \geq \max\{P_1^L + 8\epsilon + 10\delta, P_1^L + P_2^L + 6\epsilon + 6\delta, P_3^L + 4\epsilon + 4\delta\}$. For cycle S_5^3, $K \geq \max\{P_1^L + 4\epsilon + 4\delta, P_2^L + P_3^L + 6\epsilon + 6\delta, P_3^L + 8\epsilon + 10\delta\}$. In Lemma 3, we determined the optimal processing time values of the ECP for S_1^3. In the sequel we will prove similar results for the cycles S_3^3 and S_5^3, respectively.

Lemma 5 *Under cycle S_3^3, the optimal processing times of the ECP are found as follows*:

1. *If $P_1^U + P_2^U < K - 6\epsilon - 6\delta$ or $P_2^U < 2\epsilon + 4\delta$ then $P_1^* = \min\{P_1^U, K - 8\epsilon - 10\delta\}$, $P_2^* = P_2^U$, and $P_3^* = \min\{P_3^U, K - 4\epsilon - 4\delta\}$*,
2. *Otherwise, $P_3^* = \min\{P_3^U, K - 4\epsilon - 4\delta\}$, and P_1^* and P_2^* are found by solving the following two equations simultaneously: $P_1^* + P_2^* = K - 6\epsilon - 6\delta$ and $\partial f_1(P_1^*) = \partial f_2(P_2^*)$. After solving, one may get one of the following cases*:
   2.1 *If both processing times satisfy their own bounds then the solution found is optimal.*
   2.2 *Else if exactly one of the processing times, P_i^*, violates one of its bounds, P_i^b, then the optimal solution is $P_i^* = P_i^b$ and $P_j^* = K - 6\epsilon - 6\delta - P_i^b$, $i, j = 1, 2$, $i \neq j$.*
   2.3 *Else if one of the processing times (assume P_i^*) violates its lower bound (P_i^L) and the other one (P_j^*) violates its upper bound (P_j^U) then the optimal solution is found by comparing the manufacturing costs of the following two processing time settings*:
      (i) *$P_i^* = P_i^L$, $P_j^* = K - 6\epsilon - 6\delta - P_i^L$ or*
      (ii) *$P_j^* = P_j^U$, $P_i^* = K - 6\epsilon - 6\delta - P_j^U$, $i, j = 1, 2$, $i \neq j$.*

Proof In order to find the optimal processing times, the objective function of the ECP must be replaced by $f_1(P_1) + f_2(P_2) + f_3(P_3)$ and constraint (1) must be written as $\max\{P_1 + 8\epsilon + 10\delta, P_1 + P_2 + 6\epsilon + 6\delta, P_3 + 4\epsilon + 4\delta\} \leq K$.

Under this cycle, the cycle time is bounded as follows:

$$\max\{P_1^L + 8\epsilon + 10\delta, P_1^L + P_2^L + 6\epsilon + 6\delta, P_3^L + 4\epsilon + 4\delta\} \leq K$$

$$\leq \max\{P_1^U + 8\epsilon + 10\delta, P_1^U + P_2^U + 6\epsilon + 6\delta,$$
$$P_3^U + 4\epsilon + 4\delta\}. \quad (8)$$

As a result, the above constraint can be rewritten as the union of three constraints as follows:

$$P_1 + 8\epsilon + 10\delta \leq K, \quad (9)$$

$$P_1 + P_2 + 6\epsilon + 6\delta \leq K, \quad (10)$$

$$P_3 + 4\epsilon + 4\delta \leq K. \quad (11)$$

The Lagrangian for this formulation is the following:

$$\begin{aligned}
&L(P^*, \lambda^*, \mu^*)\\
&\quad = f_1(P_1^*) + f_2(P_2^*) + f_3(P_3^*)\\
&\qquad + \mu_1^*(P_1^* - K + 8\epsilon + 10\delta)\\
&\qquad + \mu_2^*(P_1^* + P_2^* - K + 6\epsilon + 6\delta)\\
&\qquad + \mu_3^*(P_3^* - K + 4\epsilon + 4\delta).
\end{aligned}$$

If we set $\nabla_P(L(P^*, \mu^*)) = 0$, we get:

$$\partial f_1(P_1^*) + \mu_1^* + \mu_2^* = 0,$$

$$\partial f_2(P_2^*) + \mu_2^* = 0,$$

$$\partial f_3(P_3^*) + \mu_3^* = 0.$$

We also have $\mu_i \geq 0$ and $P_i^L \leq P_i \leq P_i^U, \forall i$.

From the last equation we get $\mu_3^* = -\partial f_3(P_3^*)$. Since the objective is strictly convex and decreasing, $-\partial f_3(P_3^*) > 0$ unless $P_3^* = P_3^U$. Thus, constraint (11) must be satisfied as equality. However, P_3^* cannot violate its bounds. From (8), $P_3^L \leq K - 4\epsilon - 4\delta$. As a result, $P_3^* = \min\{P_3^U, K - 4\epsilon - 4\delta\}$. Now let us consider the following cases:

1. If $P_1^U + P_2^U < K - 6\epsilon - 6\delta$ or $P_2^U < 2\epsilon + 4\delta$ then constraint (10) cannot be satisfied as equality. As a result, $\mu_2^* = -\partial f_2(P_2^*) = 0 \Rightarrow P_2^* = P_2^U$. Also since $\mu_2^* = 0$, $\mu_1^* = -\partial f_1(P_1^*) > 0$ unless $P_1^* = P_1^U$. Thus, constraint (9) is satisfied as equality. However, P_3^* cannot violate its bounds. From (8), $P_1^L \leq K - 8\epsilon - 10\delta$. As a result, $P_1^* = \min\{P_1^U, K - 8\epsilon - 10\delta\}$.
2. Otherwise, $\mu_2^* = -\partial f_2(P_2^*) > 0$ unless $P_2^* = P_2^U$. Thus, in this case constraint (10) is satisfied as equality. As a result $\mu_1^* = \partial f_2(P_2^*) - \partial f_1(P_1^*) \geq 0$. Thus, we have the following cases:
   2.1. If $\partial f_1(K - 8\epsilon - 10\delta) \leq \partial f_2(2\epsilon + 4\delta)$ then $P_1^* = K - 8\epsilon - 10\delta$ and $P_2^* = 2\epsilon + 4\delta$.
   2.2. Else, solve $\partial f_1(P_1^*) = \partial f_2(P_2^*)$ and $P_1^* + P_2^* = K - 6\epsilon - 6\delta$ simultaneously to find P_1^* and P_2^*.

   Instead of these two cases we can simply represent the solution as follows:
   Solve $\partial f_1(P_1^*) = \partial f_2(P_2^*)$ and $P_1^* + P_2^* = K - 6\epsilon - 6\delta$ simultaneously to find P_1^* and P_2^*. If any of them violates its bounds, set that processing time to the violated bound and find the other one, accordingly. The upper bound for P_1^* in this case is $\min\{P_1^U, K - 8\epsilon - 10\delta\}$. □

When we compare the cycle time of this cycle with the S_5^3 cycle, we easily see that when we replace P_1 with P_3 in one of the cycle times, we get the cycle time of the other one. Thus, the analysis for these two cycles are identical. Consequently, the proof of the following lemma is very similar to the one above and will not be presented here.

Lemma 6 *Under cycle S_5^3, the optimal processing times of ECP are found as follows*:

1. *If $P_3^U + P_2^U < K - 6\epsilon - 6\delta$ or $P_2^U < 2\epsilon + 4\delta$ then $P_1^* = \min\{P_1^U, K - 8\epsilon - 10\delta\}$, $P_2^* = P_2^U$ and $P_3^* = \min\{P_3^U, K - 4\epsilon - 4\delta\}$,*
2. *Otherwise, $P_1^* = \min\{P_1^U, K - 4\epsilon - 4\delta\}$ and P_3^* and P_2^* are found by solving the following two equations simultaneously: $P_3^* + P_2^* = K - 6\epsilon - 6\delta$ and $\partial f_3(P_3^*) = \partial f_2(P_2^*)$. After solving, one may get one of the following cases*:
   2.1 *If both processing times satisfy their own bounds then the solution found is optimal.*
   2.2 *Else if exactly one of the processing times, P_i^*, violates one of its bounds, P_i^b, then the optimal solution is $P_i^* = P_i^b$ and $P_j^* = K - 6\epsilon - 6\delta - P_i^b$, $i, j = 2, 3$, $i \neq j$.*
   2.3 *Else if one of the processing times (assume P_i^*) violates its lower bound (P_i^L) and the other one (P_j^*) violates its upper bound (P_j^U) then the optimal solution is found by comparing the manufacturing costs of the following two processing time settings*:
      (i) *$P_i^* = P_i^L$, $P_j^* = K - 6\epsilon - 6\delta - P_i^L$ or*
      (ii) *$P_j^* = P_j^U$, $P_i^* = K - 6\epsilon - 6\delta - P_j^U$, $i, j = 2, 3$, $i \neq j$.*

The following is a good example to illustrate the differences of this study from the earlier ones.

Example 4 Let us consider a three-machine cell and CNC turning operations with following parameters: $\epsilon = 0.02$, $\delta = 0.1$, $T = 4$, $C_o = 0.5$, $a = -1.43423$, $U_1 = 0.2$, $U_2 = 0.03$, $U_3 = 0.75$, $P_1^L = 0.1$, $P_1^U = 1.4$, $P_2^L = 0.08$, $P_2^U = 0.64$, $P_3^L = 1.1$, $P_3^U = 2.42$. Let us determine the optimal processing times for the S_3^3 cycle with $K = 1.8$. According to Lemma 5, the optimal processing times for this cycle can be determined to be as follows: $P_1^*(S_3^3) = 0.74$, $P_2^*(S_3^3) = 0.34$, $P_3^*(S_3^3) = 1.32$. The corresponding cost for this setting of processing time is 5.012. When we calculate the cycle time of the S_6^3 cycle with this same setting of processing times, as it is the case in the current literature, we get $K = 1.8$. This means that both cycles have the same cycle time and cost values and, hence, we are indifferent between these two cycles. However, if we determine the optimal processing times for the S_6^3 cycle with $K = 1.8$ according to Lemma 4, we get $P_1^*(S_6^3) = 1.4$, $P_2^*(S_6^3) = 0.64$, $P_3^*(S_6^3) = 1.32$. The corresponding cost for this case is 4.416. This means that both cycles have the same cycle time value but the minimum cost corresponding this cycle time value for S_6^3 is less than that of S_3^3. Hence, S_3^3 is dominated.

This example shows that, with the assumptions of this study, when comparing the cycles with each other the processing time settings can be different for each cycle. Additionally, if the only criterion was the cycle time, we would conclude that both S_3^3 and S_6^3 perform equally well. However, this example proves that the cost of S_6^3 is less than S_3^3 and, hence, they can not be considered as having equal performance.

Analyzing the remaining three cycles, S_1^3, S_3^3, and S_5^3, we conclude that there is no general dominance relation among these cycles, but instead, according to the parameters such as P_i^L, P_i^U, ϵ, and δ, we can find the regions where each of them dominates the others. This is another result that differentiates this study from the earlier ones, since the decision tree provided by Sethi et al. (1992) compares all of the 1-unit cycles with each other and presents the sufficient conditions for each of them to be optimal with any given cell data, where the only objective is the minimization of the cycle time. In other words, the decision tree spans the whole feasible region. However, with the assumptions of this study, only for $K < 8\epsilon + 12\delta$ the dominance relations among the remaining three cycles depend on the cell data.

4 Different cost structures

In this section we will show how different assumptions on cost structures for the machining cost and the cost of the robot can be handled. We will present the analysis for the two-machine cells which can be extended to three-machine cells in a similar manner. The machining cost can be assumed to be either a function of the exact working time of the machines or a function of the cycle time. Till now we assumed the former of these to hold. Additionally, the cost of the robot could also be considered as an additional cost component. Although the cost incurred by the robot is relatively small in comparison with the cost incurred by the machines and the structure for the cost of the robot cannot be defined easily, we will consider two different cost structures for the robot in order to show how to handle additional cost components. In the sequel we will give insights for handling different cost structures for the machining and robot costs.

4.1 Cost allocated in terms of the cycle time

In this section we assume the machining cost to be a function of the cycle time, $C_o \cdot K$, where $K = 6\epsilon + 6\delta + P_1 + P_2$ for the S_1^2 cycle and $K = 6\epsilon + 8\delta + \max\{0, P_1 - 2\epsilon - 4\delta, P_2 - 2\epsilon - 4\delta\}$ for the S_2^2 cycle. On the other hand, in addition to the machining cost and the tooling cost, we can also consider the cost of the robot as an additional cost component. Similar to the machining cost, in this section we assume the cost of the robot to be a function of the cycle time. That is, a cost is incurred for each unit of time the cell works. The new cost terms in this case are: $R \cdot (6\epsilon + 6\delta + P_1 + P_2)$ and $R \cdot (6\epsilon + 8\delta + \max\{0, P_1 - 2\epsilon - 4\delta, P_2 - 2\epsilon - 4\delta\})\}$ for S_1^2 and S_2^2, respectively, where R represents the cost for the robot for each unit of time the cell works ($/ min).

The lower bounds for the processing times are determined by the constraints dictated by the limited tool life, the machine power, and the surface roughness. These constraints are independent of the machining and the robot costs. As a result, the lower bounds of the processing times remain unchanged. On the other hand, the upper bounds arise from the total manufacturing cost, which is different from the previous case. Furthermore, as opposed to the previous case, since the machining and robot costs for the S_1^2 and S_2^2 cycles are different from each other, the total manufacturing costs are also different leading to different upper bounds of the processing times for identical operations under these two cycles. As we mentioned earlier, the upper bound for processing time P_i is the point satisfying $\frac{\partial f_i(P_i^U)}{\partial P_i} = 0$. Note that the total manufacturing cost is assumed to be a convex function, the machining cost and the robot cost are nondecreasing functions, and the tooling cost is a nonincreasing function. We also have the following:

$$\frac{\partial((C_o + R)(6\epsilon + 6\delta + P_1 + P_2))}{\partial P_i} \geq \frac{\partial((C_o + R)(6\epsilon + 8\delta + \max\{0, P_1 - 2\epsilon - 4\delta, P_2 - 2\epsilon - 4\epsilon\}))}{\partial P_i}, \quad \forall i.$$

As a consequence, $P_i^U(S_2^2) \geq P_i^U(S_1^2)$, $\forall i$, where $P_i^U(S)$ is the upper bound of the processing time under robot move cycle S. Since the total manufacturing cost is a decreasing function of the processing time P_i for $P_i^L \leq P_i \leq P_i^U$, Lemmas 1 and 2 are still valid. Additionally, since $P_i^U(S_2^2) \geq P_i^U(S_1^2)$, $\forall i$, Theorem 1 is also valid which determines the regions where the S_1^2 and S_2^2 cycles dominate each other.

4.2 Cost allocated in terms of exact working time

Throughout this study we assumed the machining cost to be allocated in terms of the exact working time of the machines, except in Sect. 4.1. In this section we assume that the cost of the robot is also charged with respect to the exact robot activity time. That is, if R represents the unit cost for the robot activity time, then the cost incurred by the robot is $R \cdot (6\epsilon + 6\delta)$ and $R \cdot (6\epsilon + 8\delta)$ under S_1^2 and S_2^2 cycles, respectively. Note that, during an n-unit cycle, each machine is loaded and unloaded exactly n times. As a result, the total load/unload times under all cycles to produce one part are equivalent to each other. On the other hand, robot travel times differ among cycles. Comparing 1-unit cycles with each other, the robot travel time is greater under S_2^2 than under S_1^2. As a result, the cost incurred by the robot is greater under S_2^2 than S_1^2. Remember that Lemmas 1 and 2 determined the set of nondominated solutions for S_1^2 and S_2^2 cycles, respectively, where the total cost function did not include the cost of the robot. Since the cost function considered in this section for the robot is independent of the processing times on the machines, Lemmas 1 and 2 are still valid. On the other hand, without the robot cost, the remaining parts of the total cost functions for S_1^2 and S_2^2 cycles are identical, which means that for the same processing time values under both cycles, the total cost except the robot cost is equivalent for these two cycles. Since the total cost function is assumed to be decreasing for the region under consideration, the cost will not reduce with a greater processing time. This was the basic property behind the proof of Theorem 1. However, as we include the robot cost, the total cost functions for the two 1-unit robot move cycles become different from each other and Theorem 1 is no longer valid. Let $f_m(\boldsymbol{P})$ represent the machining cost and $f_t(\boldsymbol{P})$ represent the tooling cost with processing time vector $\boldsymbol{P}$. The new breakpoint for the region of dominance satisfies the following:

$$\begin{aligned}
\bar{K} &= 6\epsilon + 6\delta + \hat{P}_1 + \hat{P}_2 \\
&= 6\epsilon + 8\delta + \max\{0, \tilde{P}_1 - 2\epsilon - 4\delta, \tilde{P}_2 - 2\epsilon - 4\delta\}, \\
f_m(\hat{P}_1, \hat{P}_2) &+ f_t(\hat{P}_1, \hat{P}_2) + R \cdot (6\epsilon + 6\delta) \\
&= f_m(\tilde{P}_1, \tilde{P}_2) + f_t(\tilde{P}_1, \tilde{P}_2) + R \cdot (6\epsilon + 8\delta)
\end{aligned}$$

where $\hat{P}_i \in P_i^*(S_1^2)$ and $\tilde{P}_i \in P_i^*(S_2^2)$, $i = 1, 2$. If $K \le \bar{K}$ then, $S_1^2 \preceq S_2^2$; otherwise, $S_2^2 \preceq S_1^2$.

The following is an example showing that the breakpoint found in Theorem 1 is not valid for this new situation.

Example 5 Let us consider the turning operation for which the total cost function for the S_1^2 cycle can be written as $C_o \cdot (P_1 + P_2) + T_1U_1P_1^{a_1} + T_2U_2P_2^{a_2} + R \cdot (6\epsilon + 6\delta)$. The only difference of the cost function for the S_2^2 cycle is the robot cost which is $R \cdot (6\epsilon + 8\delta)$ for S_2^2. Let $T_1 = 6$, $T_2 = 5$, $C_o = 0.9$, $R = 2$, $\epsilon = 0.1$, $\delta = 0.2$, $U_1 = 0.03$, $U_2 = 0.8$, $a_1 = -0.6$, and $a_2 = -1$. According to the given parameters, the upper bounds for the processing times are found to be $P_1^U = 0.266$ and $P_2^U = 2.108$. Let $K = 3.3 > 6\epsilon + 8\delta = 2.2$. Then, if the robot cost is ignored, for $K = 3.3$, according to Theorem 1, $S_2^2 \preceq S_1^2$. With the inclusion of the robot cost, let $\hat{P}_1 = 0.152$, $\hat{P}_2 = 1.348$, where $\hat{P}_i \in P^*(S_1^2)$ and $\tilde{P}_1 = 0.266$, $\tilde{P}_2 = 2.1$, where $\tilde{P}_i \in P^*(S_2^2)$. As a consequence, $F_2(S_1^2, (0.152, 1.348)) = F_2(S_2^2, (0.237, 2.1)) = 3.3$. On the other hand, $F_1(S_1^2, (0.152, 1.348)) = 8.475$ and $F_1(S_2^2, (0.237, 2.1)) = 8.832$. As a result, in contrast to Theorem 1, with the presence of robot cost $S_1^2 \preceq S_2^2$, even though $K > 6\epsilon + 8\delta$.

This example shows that considering the cycle time as the only performance measure hinders the other characteristics of the solutions. Although a solution may have a small cycle time value, it may be dominated because of its poor cost performance. Even the basic results of Sethi et al. (1992) regarding the two-machine identical parts robotic cell scheduling problem are not valid when the cost is considered as a performance measure simultaneously with the cycle time. This brings additional insights to the problem and provides flexibility for the decision maker by determining the set of efficient solutions.

5 Conclusion

In this study we considered robotic cell scheduling with identical parts in two- and three-machine robotic cells. The machines in a robotic cell used in metal cutting industries are predominantly CNC machines so that the machines and the robot can interact on a real time basis by the help of the cell controller. These machines are highly flexible. The processing times of the parts on these machines can be controlled by adjusting the machining conditions such as the speed and the feed rate. However, adjusting these parameters also affects the tool life which consequently affects the total manufacturing cost. Hence, in this study we considered a bicriteria robotic cell scheduling problem in which the robot move sequence as well as the processing times on the machines are the decision variables and the cycle time and the total manufacturing cost are the performance measures. Since there are two competing performance measures, instead of a unique optimal solution a set of nondominated solutions exists for such problems.

We determined the set of nondominated solutions for the two 1-unit cycles of two-machine robotic cells in Lemmas 1 and 2, and compared these two cycles with each other in Theorem 1. A similar analysis is performed for three-machine cells also. Theorem 3 proves that two of the six 1-unit cycles of a three-machine cell are dominated and need not be considered. Lemmas 3, 4, 5, and 6 determine the nondominated set of solutions for the remaining four cycles. By comparing these with each other, Theorem 4 determines the regions where S_6^3 dominates the rest. Note that no dominance relations exist between the remaining three cycles for the remaining very small region. We made our analysis for any strictly convex, differentiable cost function. In Sect. 4 we showed how different assumptions on cost structures can be handled. These assumptions include changing the allocation base for the machining cost and including the cost incurred by the robot within the analysis. We showed that if the machining cost and the cost of the robot are allocated with respect to the cycle time, earlier results found in this study are still valid. However, if the robot cost is allocated with respect to the exact working time of the robot, the regions of optimality for the S_1^2 and S_2^2 cycles change. This change is in favor of the S_1^2 cycle under which number of robot moves is less than the number of robot moves under S_2^2 cycle.

As far as the authors know, this is the first study to consider cost objectives in robotic cell scheduling literature. As a future research direction, the results of this study may be extended to m-machine cells or cells producing multiple parts. The complexity of the problem increases drastically in both cases, since the number of feasible 1-unit cycles in an m-machine cell is exactly $m!$ and the sequencing of the parts is also a decision problem for cells producing multiple parts. In this study we assumed that each part has one operation to be performed on each machine. Another future research direction is to extend the analysis of this study so that not only 1-unit cycles but all feasible cycles are considered as alternatives.

Appendix

Here we will present the robot activity sequences and the cycle times of the six feasible 1-unit cycles for a three-machine robotic cell.

S_1^3: $A_0A_1A_2A_3$: $8\epsilon + 8\delta + P_1 + P_2 + P_3$,

S_2^3: $A_0A_2A_1A_3$:

$$\max\{8\epsilon + 12\delta, P_1 + 6\epsilon + 8\delta, P_2 + 4\epsilon + 4\delta, P_3 + 6\epsilon + 8\delta, (P_1 + P_2 + P_3)/2 + 4\epsilon + 4\delta\},$$

S_3^3: $A_0A_1A_3A_2$:

$$\max\{P_1 + 8\epsilon + 10\delta, P_1 + P_2 + 6\epsilon + 6\delta, P_3 + 4\epsilon + 4\delta\},$$

S_4^3: $A_0A_3A_1A_2$:

$$\max\{P_1 + P_2 + 6\epsilon + 6\delta, P_2 + 8\epsilon + 12\delta, P_2 + P_3 + 6\epsilon + 6\delta\},$$

S_5^3: $A_0A_2A_3A_1$:

$$\max\{P_1 + 4\epsilon + 4\delta, P_2 + P_3 + 6\epsilon + 6\delta, P_3 + 8\epsilon + 10\delta\},$$

S_6^3: $A_0A_3A_2A_1$:

$$\max\{8\epsilon + 12\delta, P_1 + 4\epsilon + 4\delta, P_2 + 4\epsilon + 4\delta, P_3 + 4\epsilon + 4\delta\}.$$

References

Akturk, M. S., Gultekin, H., & Karasan, O. E. (2005). Robotic cell scheduling with operational flexibility. *Discrete Applied Mathematics*, *145*(3), 334–348.

Brauner, N., & Finke, G. (1999). On the conjecture in robotic cells: new simplified proof for the three-machine case. *INFOR: Information Systems and Operational Research*, *37*, 20–36.

Cheng, T. C. E., Janiak, A., & Kovalyov, M. Y. (1998). Bicriterion single machine scheduling with resource dependent processing times. *SIAM Journal on Optimization*, *8*, 617–630.

Chu, C. (2006). A faster polynomial algorithm for 2-cyclic robotic scheduling. *Journal of Scheduling*, *9*, 453–468.

Crama, Y., & Van de Klundert, J. (1997). Cyclic scheduling of identical parts in a robotic cell. *Operations Research*, *45*(6), 952–965.

Crama, Y., & Van de Klundert, J. (1999). Cyclic scheduling in 3-machine robotic flow shops. *Journal of Scheduling*, *4*, 35–54.

Crama, Y., Kats, V., Van de Klundert, J., & Levner, E. (2000). Cyclic scheduling in robotic flowshops. *Annals of Operations Research*, *96*, 97–124.

Daniels, R. L., & Sarin, R. K. (1989). Single machine scheduling with controllable processing times and number of jobs tardy. *Operations Research*, *37*, 981–984.

Dawande, M., Geismar, H. N., Sethi, S., & Sriskandarajah, C. (2005). Sequencing and scheduling in robotic cells: recent developments. *Journal of Scheduling*, *8*(5), 387–426.

Hall, N. G., Kamoun, H., & Sriskandarajah, C. (1997). Scheduling in robotic cells: classification, two and three machine cells. *Operations Research*, *45*(3), 421–439.

Hoogeveen, H. (2005). Multicriteria Scheduling. *European Journal of Operational Research*, *167*, 592–623.

Janiak, A., & Kovalyov, M. Y. (1996). Single machine scheduling subject to deadlines and resource dependent processing times. *European Journal of Operational Research*, *94*, 284–291.

Kayan, R. K., & Akturk, M. S. (2005). A new bounding mechanism for the CNC machine scheduling problems with controllable processing times. *European Journal Operational Research*, *167*, 624–643.

Lee, C. Y., Lei, L., & Pinedo, M. (1997). Current trends in deterministic scheduling. *Annals of Operations Research*, *70*, 1–41.

Nowicki, E., & Zdrzalka, S. (1995). A survey of results for sequencing problems with controllable processing times. *Discrete Applied Mathematics*, *63*, 237–256.

Sethi, S. P., Sriskandarajah, C., Sorger, G., Blazewicz, J., & Kubiak, W. (1992). Sequencing of parts and robot moves in a robotic cell. *International Journal of Flexible Manufacturing Systems*, *4*, 331–358.

Shabtay, D., & Kaspi, M. (2004). Minimizing the total weighted flow time in a single machine with controllable processing times. *Computers & Operations Research*, *31*, 2279–2289.

T'kindt, V., & Billaut, J.-C. (2006). *Multicriteria scheduling: theory, models and algorithms* (2nd ed.). Berlin: Springer.

Vickson, R. G. (1980). Choosing the job sequence and processing times to minimize processing plus flow cost on a single machine. *Operations Research*, *28*, 1155–1167.

van Wassenhove, L., & Baker, K. R. (1982). A bicriterion approach to time/cost trade-offs in sequencing. *European Journal of Operational Research*, *11*, 48–54.