Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem*

THOMAS DUNKER[†], GÜNTER RADONS[‡] AND ENGELBERT WESTKÄMPER[§]

Abstract. *This paper presents an algorithm combining dynamic programming and genetic search for solving a dynamic facility layout problem. While the quadratic assignment formulation of this problem has been deeply investigated there are very few papers solving it for departments of unequal size. We describe a model which can cope with an unequal and changing size. The genetic algorithm evolves a population of layouts for each time period while the dynamic programming provides the evaluation of the fitness of the layouts.*

Key words: *genetic algorithms, dynamic programming, dynamic facility layout problem*

1. Introduction

One task in factory planning is to determine good locations for a given set of departments on some workshop floor - frequently called facility layout problem (FLP). A first objective is to minimize material handling costs. Yet, other criteria like manpower requirements, work-in-process inventory, flow of information, etc. may play an important role, too. In all cases there are proximities between certain departments which are more favorable than others.

There are industrial production processes which require rigid installation of heavy equipment. The layout of such a plant has to ensure good system performance for a wide variety of demand scenarios. Yet, many products can be processed on flexible manufacturing systems which allow to respond efficiently to changes in production demands by changing the layout. In this case the layout designer is faced with the problem to consider carefully the costs of rearranging a layout and the savings by an improved performance. The first algorithmic treatment of this problem

*This research was supported by DFG research project SFB 467 "Wandlungsfähige Unternehmensstrukturen für die variantenreiche Serienproduktion"

†Fraunhofer Institute of Manufacturing Engineering and Automation, Nobelstr. 12, 70 569 Stuttgart, Germany, e-mail: tmd@ipa.fhg.de

‡Technical University Chemnitz, Institute of Physics, Reichenhainer Str. 70, 09 126 Chemnitz, Germany

§Fraunhofer Institute of Manufacturing Engineering and Automation, Nobelstr. 12, 70 569 Stuttgart, Germany

dates back to the work of Rosenblatt [17] who assumed equal size departments and formulated the problem as a dynamic program. We present an approach which can handle unequal and changing sizes of departments. Our algorithm combines genetic search (GS) and dynamic programming (DP).

This paper is organized as follows. In *Section 2* we introduce a mathematical model for the dynamic facility layout problem (DFLP) for unequally sized departments. After reviewing different solution methods for DFLPs we present our algorithm in *Section 3*. Finally, we discuss our numerical experiments and compare them with other works in *Section 4*.

2. The model

As in Yang and Peters [19] we model workshop floor area and departments by rectangles. We denote them by $A_i^{(t)}$, where $i, j \in I = \{1, \ldots, N\}$ serve as indices to distinguish different rectangles and $t \in \{0, \ldots, T\}$ stands for the considered period of time. Let $A_0^{(t)}$ be the the total available floor area which does not change. The shape of a rectangle is determined by the side lengths $l_i^{(t)}$ and $s_i^{(t)}$ which can change from period to period. This allows us to model expansion, shrinking or replacement by a new department. The position of a rectangle can be described by its center point $(X_i^{(t)}, Y_i^{(t)})$ and orientation $O_i^{(t)} \in \{0,1\}$ (vertical/horizontal).

In order to ensure that department $A_i^{(t)}$ is placed inside the total available floor area we introduce the following four constraints

$$X_0^{(t)} - \frac{l_0^{(t)}}{2} O_0^{(t)} - \frac{s_0^{(t)}}{2}(1 - O_0^{(t)}) \leq X_i^{(t)} - \frac{l_i^{(t)}}{2} O_i^{(t)} - \frac{s_i^{(t)}}{2}(1 - O_i^{(t)}), \quad (1)$$

$$X_i^{(t)} + \frac{l_i^{(t)}}{2} O_i^{(t)} + \frac{s_i^{(t)}}{2}(1 - O_i^{(t)}) \leq X_0^{(t)} + \frac{l_0^{(t)}}{2} O_0^{(t)} + \frac{s_0^{(t)}}{2}(1 - O_0^{(t)}), \quad (2)$$

$$Y_0^{(t)} - \frac{s_0^{(t)}}{2} O_0^{(t)} - \frac{l_0^{(t)}}{2}(1 - O_0^{(t)}) \leq Y_i^{(t)} - \frac{s_i^{(t)}}{2} O_i^{(t)} - \frac{l_i^{(t)}}{2}(1 - O_i^{(t)}), \quad (3)$$

$$Y_i^{(t)} + \frac{s_i^{(t)}}{2} O_i^{(t)} + \frac{l_i^{(t)}}{2}(1 - O_i^{(t)}) \leq Y_0^{(t)} + \frac{s_0^{(t)}}{2} O_0^{(t)} + \frac{l_0^{(t)}}{2}(1 - O_0^{(t)}). \quad (4)$$

The part of the model which makes it so hard for the numerical treatment is the non-overlapping of two rectangles which requires the existence of a vertical or horizontal line which separates the two rectangles. We propose a model using two variables. We introduce variables $S_{ij}^{D\,(t)} \in \{0,1\}$ for the direction of the separating line and $S_{ij}^{O\,(t)} \in \{0,1\}$ for the order of $A_i^{(t)}$ and $A_j^{(t)}$, with $i, j \in I$. In order to avoid redundant variables we require $i > j$. Let $l_{\max} = \max_{i,t} l_i^{(t)}$. Using the four linear expressions

$$E_{ij}^{1\,(t)} = S_{ij}^{D\,(t)} + S_{ij}^{O\,(t)}, \quad (5)$$

$$E_{ij}^{2\,(t)} = 1 + S_{ij}^{D\,(t)} - S_{ij}^{O\,(t)}, \quad (6)$$

$$E_{ij}^{3\,(t)} = 1 - S_{ij}^{D\,(t)} + S_{ij}^{O\,(t)}, \quad (7)$$

$$E_{ij}^{4\,(t)} = 2 - S_{ij}^{D\,(t)} - S_{ij}^{O\,(t)} \tag{8}$$

where each takes the value zero for exactly one setting of $(S_{ij}^{D\,(t)}, S_{ij}^{O\,(t)})$ we can derive the necessary constraints

$$X_i^{(t)} + \frac{l_i^{(t)}}{2} O_i^{(t)} + \frac{s_i^{(t)}}{2}(1 - O_i^{(t)}) - l_{\max} E_{ij}^{1\,(t)} \le X_j^{(t)} - \frac{l_j^{(t)}}{2} O_j^{(t)} - \frac{s_j^{(t)}}{2}(1 - O_j^{(t)}), \tag{9}$$

$$X_j^{(t)} + \frac{l_j^{(t)}}{2} O_j^{(t)} + \frac{s_j^{(t)}}{2}(1 - O_j^{(t)}) - l_{\max} E_{ij}^{2\,(t)} \le X_i^{(t)} - \frac{l_i^{(t)}}{2} O_i^{(t)} - \frac{s_i^{(t)}}{2}(1 - O_i^{(t)}), \tag{10}$$

$$Y_i^{(t)} + \frac{s_i^{(t)}}{2} O_i^{(t)} + \frac{l_i^{(t)}}{2}(1 - O_i^{(t)}) - l_{\max} E_{ij}^{3\,(t)} \le Y_j^{(t)} - \frac{s_j^{(t)}}{2} O_j^{(t)} - \frac{l_j^{(t)}}{2}(1 - O_j^{(t)}), \tag{11}$$

$$Y_j^{(t)} + \frac{s_j^{(t)}}{2} O_j^{(t)} + \frac{l_j^{(t)}}{2}(1 - O_j^{(t)}) - l_{\max} E_{ij}^{4\,(t)} \le Y_i^{(t)} - \frac{s_i^{(t)}}{2} O_i^{(t)} - \frac{l_i^{(t)}}{2}(1 - O_i^{(t)}). \tag{12}$$

The performance of a layout is determined by the distances between drop off, pick up and other points of departments and floor area. We will call these points IO-points. For simplicity we consider the case that the IO-points are approximated by the center points of the departments. In addition, there might be IO-points $(X_i^{(t)}, Y_i^{(t)})$, $i = N+1, \ldots, N+k$, fixed to the floor area and we define $I^{\bullet} = \{1, \ldots, N+k\}$

Furthermore, we need rectilinear distances $\delta X_{ij}^{(t)}, \delta Y_{ij}^{(t)}$ between IO-points and weights $\left(w_{ij}^{(t)}\right)_{\substack{i,j \in I^{\bullet} \\ i>j}}$ which represent e.g. material handling cost per meter. Next, we introduce variables $M_i^{(t)} \in \{0,1\}$ for $i \in I$ which take value 1 if there has been a change of position from $A_i^{(t-1)}$ to $A_i^{(t)}$. Weights $\left(w_i^{(t)}\right)_{i \in I}$ represent e.g. costs for moving departments. Denote by $\mathbb{L}^{(t)}$ the set of all possible layouts $L_t = \left(X_i^{(t)}, Y_i^{(t)}, O_i^{(t)}\right)_i$ in period t. With this notation the objective is to minimize the following expression

$$C^*(L_0, 0, T) := \min \sum_{t=1}^{T} \Bigg[\underbrace{\sum_{\substack{i,j \in I^{\bullet} \\ i>j}} w_{ij}^{(t)} (\delta X_{ij}^{(t)} + \delta Y_{ij}^{(t)})}_{=:\, F(L_t, t)} + \underbrace{\sum_{i \in I} w_i^{(t)} M_i^{(t)}}_{=:\, A(L_{t-1}, L_t, t)} \Bigg] \tag{13}$$

for some fixed $L_0 \in \mathbb{L}^{(0)}$ where the following constraints have to be satisfied

$$X_i^{(t)} - X_j^{(t)} \le \delta X_{ij}^{(t)} \quad \text{and} \quad X_j^{(t)} - X_i^{(t)} \le \delta X_{ij}^{(t)}, \tag{14}$$

$$Y_i^{(t)} - Y_j^{(t)} \le \delta Y_{ij}^{(t)} \quad \text{and} \quad Y_j^{(t)} - Y_i^{(t)} \le \delta Y_{ij}^{(t)}, \tag{15}$$

for all $i, j \in I^{\bullet}$ with $i > j$ and $w_{ij}^{(t)} \neq 0$ and

$$X_i^{(t)} - X_i^{(t-1)} \le l_{\max} M_i^{(t)} \quad \text{and} \quad X_i^{(t-1)} - X_i^{(t)} \le l_{\max} M_i^{(t)}, \tag{16}$$

$$Y_i^{(t)} - Y_i^{(t-1)} \le l_{\max} M_i^{(t)} \quad \text{and} \quad Y_i^{(t-1)} - Y_i^{(t)} \le l_{\max} M_i^{(t)}, \tag{17}$$

$$O_i^{(t)} - O_i^{(t-1)} \le M_i^{(t)} \quad \text{and} \quad O_i^{(t-1)} - O_i^{(t)} \le M_i^{(t)} \tag{18}$$

for all $i \in I$. Thus our mixed integer program (MIP) is completely described by equations (1) – (18). All equalities, inequalities and the objective are linear. The model contains binary variables $O_i^{(t)}$, $M_i^{(t)}$, $S_{ij}^{D^{(t)}}$ and $S_{ij}^{O^{(t)}}$. While the number of the first ones increases proportionally to N and T, the number of $S_{ij}^{D^{(t)}}$'s and $S_{ij}^{O^{(t)}}$'s increases quadratically with N.

3. The algorithm

Let us consider the DFLP with a finite number of locations for placing the departments. In this case sets $\mathbb{L}^{(t)}$ are finite. Rosenblatt [17] first solved this problem by DP which is based on the following recursive relationship

$$C^*(L_t, t, T) = \min_{L \in \mathbb{L}^{(t+1)}} \{A(L_t, L, t+1) + C^*(L, t+1, T)\} + F(L_t, t). \tag{19}$$

As $\#\mathbb{L}^{(t)} \geq N!$ increases rapidly with the number of departments, the size of problems which can be solved by this dynamic program is very limited. Hence there has been much effort to restrict the dynamic program to "good" subsets of the $\mathbb{L}^{(t)}$'s, see Balakrishnan and Cheng [1], *Section 4.1*, for a survey of different methods and further references. Balakrishnan et al. [3] extended the problem by budgeting the rearrangement cost and use instead of DP a constrained shortest path algorithm proposed by Mote et al. [16].

A second approach to the dynamic facility layout is to reduce it to a problem with a fixed initial layout and a single period (1-PFLP), see e.g. problem formulation in Kochhar and Heragu [12]. Having a layout for a preceding period one solves the 1-PFLP obtaining a layout for the next period. In addition one can aggregate several periods to a 1-PFLP. This way one prohibits rearrangements during this long period. Then one can combine aggregations and the 1-PFLP method and apply it to all 2^T partitions of the sequence $1, \ldots, T$. In case of the so-called fixed rearrangement cost problem (the cost depends only on the fact whether there are rearrangements or not) this method – which needs the evaluation of $T(T-1)/2$ FLPs – gives optimal solutions, see Urban [18] and for a similar method Balakrishnan et al. [4]. Yang and Peters [19] propose a special strategy for finding the periods to be aggregated to a long one. They add one period at the time as long as the cost per period of the solution decreases. See Balakrishnan and Cheng [1], *Section 4.1*, for further improvement methods and references.

In order to cope with rapidly increasing complexity of the problem many authors propose a heuristic algorithm to obtain "good" solutions, see e.g. Kaku and Mazzola [11] (tabu search), Hirabayshi et al. [10] (evolution strategies) or Baykasoglu and Gindy [5] (simulated annealing). There are numerous algorithms using GS, see e.g. Conway and Venkataramanan [8], Coit et al. [7], Kochhar and Heragu [12] or Balakrishnan and Cheng [2].

In the field of problems with unequal and changing department size there are less publications. Montreuil and Venkatadri [15] and Montreuil and Laforge [14] propose algorithms where the designer has to fix relative positions between the departments in advance — the so-called skeleton. A two stage algorithm developed by Lacksonen

[13] automatizes this first step. Yet, the skeleton restricts very much the possible rearrangements. Yang and Peters [19] overcome this shortcoming by the above mentioned strategy and an algorithm from Goetschalckx [9] for the single period problem. For flexible bay layouts — a special case of an unequal and changing size problem — Coit et al. [7] propose a GA.

Our approach combines DP and GS. For each period t we have population $\mathbb{P}_\gamma^{(t)} = \{\mathbb{I}_{\gamma 1}^{(t)}, \ldots, \mathbb{I}_{\gamma N^{(t)}}^{(t)}\}$ of $N^{(t)}$ individuals representing different layouts $L_{\gamma\iota}^{(t)}$. The index $\gamma = 1, 2, \ldots$ stands for "generation" and a genetic algorithm (GA) generates $\mathbb{P}_{\gamma+1}^{(t)}$ from $\mathbb{P}_\gamma^{(t)}$. In order to be able to apply a GA we need a fitness value for each individual. For this purpose we determine for each $\mathbb{I}_{\gamma\iota}^{(t)}$ the minimal cost over all sequences $L_0, L_{\gamma\iota_1}^{(1)}, \ldots, L_{\gamma\iota}^{(t)}, \ldots, L_{\gamma\iota_T}^{(T)}$ of layouts where $L_{\gamma\iota_s}^{(s)}$ corresponds to $\mathbb{I}_{\gamma\iota_s}^{(s)} \in \mathbb{P}_\gamma^{(s)}$. This can easily be done by forward and backward DP.

3.1. Coding and genetic operators

The ι-th individual of population $\mathbb{P}_\gamma^{(t)}$ will be represented by

$$\mathbb{I}_{\gamma\iota}^{(t)} = \Big(\underbrace{\big(\{n_i^x\}_{i\in I}, \{n_i^y\}_{i\in I}, \{b_{ij}\}_{\substack{i,j\in I\\ i>j}}\big)}_{\text{genotype}}, \underbrace{\big(\{x_i\}_{i\in I}, \{y_i\}_{i\in I}, \{o_i\}_{i\in I}\big)}_{\text{phenotype}}\Big).$$

The element $b_{ij} \in \{0,1\}$ represents a value of the binary variable $S_{ij}^{D(t)}$. The two vectors $(n_1^x, \ldots, n_N^x)$ and $(n_1^y, \ldots, n_N^y)$ are permutations of the elements of I and they represent the order of the x- and y-coordinates of the center points of the rectangles. The phenotype part stores values for the position variables $X_i^{(t)}$, $Y_i^{(t)}$ and $O_i^{(t)}$ which are obtained as described below.

Given an $\mathbb{I}_{\gamma\iota}^{(t)}$ we set the $S_{ij}^{D(t)}$ and $S_{ij}^{O(t)}$ with $i, j \in I$ and $i > j$ in the following way. For each pair of indices $1 \le j_1 < j_2 \le N$ we check the following alternatives. Set $i^x = \max(n_{j_1}^x, n_{j_2}^x)$ and $j^x = \min(n_{j_1}^y, n_{j_2}^y)$. If $b_{i^x j^x} = 0$ holds, then set

$$S_{i^x j^x}^{D}{}^{(t)} = 0 \text{ and } S_{i^x j^x}^{O}{}^{(t)} = \begin{cases} 0 & \text{if } i^x = n_{j_1}^x \\ 1 & \text{if } i^x = n_{j_2}^x \end{cases}. \tag{20}$$

Note that if $b_{i^x j^x} = 1$ the order $n_{j_1}^x$, $n_{j_2}^x$ does not enter in problem formulation. Hence the final order of the x-coordinates of $A_{n_{j_1}^x}^{(t)}$ and $A_{n_{j_2}^x}^{(t)}$ is not necessarily $X_{n_{j_1}^x}^{(t)} \le X_{n_{j_2}^x}^{(t)}$. Analogously, for the y-direction we set $i^y = \max(n_{j_1}^y, n_{j_2}^y)$ and $j^y = \min(n_{j_1}^y, n_{j_2}^y)$. If $b_{i^y j^y} = 1$ is true, then we assign

$$S_{i^y j^y}^{D}{}^{(t)} = 1 \text{ and } S_{i^y j^y}^{O}{}^{(t)} = \begin{cases} 0 & \text{if } i^y = n_{j_1}^y \\ 1 & \text{if } i^y = n_{j_2}^y \end{cases}. \tag{21}$$

Given a second individual $\mathbb{I}_{\gamma\iota'}^{(t-1)} \in \mathbb{P}_\gamma^{(t-1)}$ variables $X_i^{(t-1)}$, $Y_i^{(t-1)}$ and $O_i^{(t-1)}$ are fixed to the values x_i, y_i and o_i. We drop all variables not belonging to period t and $t-1$ from our MIP. This way we obtain a model for a 1-PFLP where the spatial relations between the rectangles are fixed. The number of the remaining binary variables increases linearly with the number of departments and problems with some dozens of departments do not cause difficulties to solvers like CPLEX. After solving this 1-PFLP we store the values of $X_i^{(t)}$'s, $Y_i^{(t)}$'s and $O_i^{(t)}$'s in the phenotype part.

The genetic operators crossover and mutation act on the genotype part. As the phenotype is the solution of a 1-PFLP we need always an individual $\mathbb{I}_{\gamma\iota'}^{(t-1)}$ providing an initial layout. Our crossover is a version of the order crossover presented in Chan and Tansri [6]. After selecting $\mathbb{I}_{\gamma\iota'}^{(t-1)}$ providing an initial layout and two parent genes $\mathbb{I}_{\gamma\iota_1}^{(t)}$ and $\mathbb{I}_{\gamma\iota_2}^{(t)}$ let us consider the parts of the genes representing the x- and y-order. Take e.g.

$$(n_{1,\iota_1}^x, \ldots, n_{N,\iota_1}^x) \quad \text{and} \quad (n_{1,\iota_2}^x, \ldots, n_{N,\iota_2}^x)$$

where we add the index of the individual as a second subindex. We select randomly two cut positions $c_1, c_2 \in \{1, \ldots, N\}$ with $c_1 \leq c_2$. Then we construct two new genes from the two selected genes. First we fill the position from c_1 to c_2 with the original parts of the sequence

$$(\ldots, n_{c_1,\iota_1}^x, \ldots, n_{c_2,\iota_1}^x, \ldots) \quad \text{and} \quad (\ldots, n_{c_1,\iota_2}^x, \ldots, n_{c_2,\iota_2}^x, \ldots).$$

Then the position before c_1 and after c_2 are filled with with the numbers from the other parent which are not contained in the already filled part. While filling we keep the order given by the parent where we take the elements from. In terms of the notation above this means e.g. for the first offspring gene

$$(n_{f(1),\iota_2}^x, \ldots, n_{f(c_1-1),\iota_2}^x, n_{c_1,\iota_1}^x, \ldots, n_{c_2,\iota_1}^x, n_{f(c_1),\iota_2}^x, \ldots, n_{f(N-c_2+c_1-1),\iota_2}^x) \quad \text{with}$$

$$\{n_{f(1),\iota_2}^x, \ldots, n_{f(N-c_2+c_1-1),\iota_2}^x\} \cap \{n_{c_1,\iota_1}^x, \ldots, n_{c_2,\iota_1}^x\} = \varnothing$$

and the mapping $j \mapsto f(j)$ is strictly increasing. In the same way the crossover is defined for the y-direction. The motivation for this definition is the idea that we wish to keep the part that is located between the cuts hoping that it contributes to a good solution and arranging the remaining using the order given by the other parent. The mutation operator on $\{n_i^x\}_{i\in I}$ and $\{n_i^y\}_{i\in I}$ exchanges just two randomly chosen elements.

The more complicated part is the modification of $\{b_{ij}\}_{i,j\in I, i>j}$ which represents the decision whether to have a vertical or a horizontal separating line. As we did not find a method which could be motivated geometrically we decided to use a standard crossover with two cut positions and standard mutation. Yet, in addition we apply an improvement strategy. First, we fix variables $S_{ij}^{D^{(t)}}$ and $S_{ij}^{O^{(t)}}$ for the given individual $\mathbb{I}_{\gamma\iota}^{(t)}$ according to (20) – (21), drop all variables not belonging to period t and $t-1$ and solve the remaining 1-PFLP with fixed spatial relations.

```
evaluate (I_{γι}^{(t)})
    for all i,j ∈ I with i > j
        fix S_{ij}^{D(t)} and S_{ij}^{O(t)}
    endfor
    solve remaining MIP for 1-PFLP
    return solution
```

Next, we update $\{n_i^x\}_{i\in I}$ and $\{n_i^y\}_{i\in I}$ by sorting the center point coordinates of the obtained solution. Then we check for all $S_{ij}^{D(t)}$ whether they can be changed without violating a constraint in the current solution.

```
change_is_possible (S_{ij}^{D(t)})
    if S_{ij}^{D(t)} == 0 (x-direction)
        if |Y_i^{(t)} - Y_j^{(t)}| ≥ s_i^{(t)}O_i^{(t)}/2 + l_i^{(t)}(1 - O_i^{(t)})/2 + s_j^{(t)}O_j^{(t)}/2 + l_j^{(t)}(1 - O_j^{(t)})/2
            return true
        else
            return false
        endif
    else (y-direction)
        if |X_i^{(t)} - X_j^{(t)}| ≥ l_i^{(t)}O_i^{(t)}/2 + s_i^{(t)}(1 - O_i^{(t)})/2 + l_j^{(t)}O_j^{(t)}/2 + s_j^{(t)}(1 - O_j^{(t)})/2
            return true
        else
            return false
        endif
    endif
```

If it is possible we change variable $S_{ij}^{D(t)}$. These actions are repeated until the objective value does not decrease further. Summarizing we have sketched our improvement strategy below.

```
improve (I_{γι}^{(t)})
    while the objective value decreases
        evaluate (I_{γι}^{(t)})
        obtain {n_i^x}_{i∈I} and {n_i^y}_{i∈I} from the solution
        for all i,j ∈ I with i > j
            if change_is_possible (S_{ij}^{D(t)})
                if S_{ij}^{D(t)} == 0
                    b_{ij} = 1
                else
                    b_{ij} = 0
                endif
            endif
        endfor
    endwhile
    return I_{γι}^{(t)} with smallest objective value
```

3.2. Description of the dynamic GA

As already sketched, the evaluation of the fitness of all individuals is obtained by DP. Let us denote by $C_{\gamma\iota}^{(t)}$ the fitness value for $\mathbb{I}_{\gamma\iota}^{(t)}$. As auxiliary variables we need $C_{\gamma\iota}^{(t)-}$ and $C_{\gamma\iota}^{(t)+}$ representing the minimal cost of a sequence of layouts from the given populations leading to layout $L_{\gamma\iota}^{(t)}$ and starting from it, respectively. The DP can be summarized:

dynamic_program($\mathbb{P}_\gamma^{(1)}, \ldots, \mathbb{P}_\gamma^{(T)}$)
  for all periods t from 1 to T
    for all $\iota = 1, \ldots, N^{(t)}$

$$C_{\gamma\iota}^{(t)-} = \min_{\iota'=1,\ldots,N^{(t-1)}} \left\{ C_{\gamma\iota'}^{(t-1)-} + F(L_{\gamma\iota'}^{(t-1)}, t-1) + A(L_{\gamma\iota'}^{(t-1)}, L_{\gamma\iota}^{(t)}, t) \right\}$$

    endfor
  endfor
  for all periods t from T to 1
    for all $\iota = 1, \ldots, N^{(t)}$

$$C_{\gamma\iota}^{(t)+} = \min_{\iota'=1,\ldots,N^{(t+1)}} \left\{ C_{\gamma\iota'}^{(t+1)+} + F(L_{\gamma\iota'}^{(t+1)}, t+1) + A(L_{\gamma\iota}^{(t)}, L_{\gamma\iota'}^{(t+1)}, t+1) \right\}$$

$$C_{\gamma\iota}^{(t)} = C_{\gamma\iota}^{(t)-} + F(L_{\gamma\iota}^{(t)}, t) + C_{\gamma\iota}^{(t)+}$$

    endfor
  endfor

Let us introduce some further parameters for the GA. We denote by N^{cr} the number of crossover operations. N^{mu} stands for the number of mutations. Furthermore, we allow migration of the best individuals from $\mathbb{P}_{\gamma-1}^{(t-1)}$, $\mathbb{P}_{\gamma-1}^{(t)}$ and $\mathbb{P}_{\gamma-1}^{(\min(t+1,T))}$ to $\mathbb{P}_\gamma^{(t)}$. Let $N^{\mathsf{im}-}$, N^{im} and $N^{\mathsf{im}+}$ denote the respective numbers of migrating individuals. Hence the size $N^{(t)}$ of each population is $2N^{\mathsf{cr}} + N^{\mathsf{mu}} + N^{\mathsf{im}-} + N^{\mathsf{im}} + N^{\mathsf{im}+}$. For stopping the algorithm we consider the average of all fitness values. M^{ch} denotes the minimal change rate required for the average. Let N^{nc} be the maximal allowed number of generations without change of the best fitness value. Moreover, we denote by M^{ge} the maximal number of generations which we allow. Then our algorithm works as follows.

dynamic_genetic_algorithm
  initialize the populations $\mathbb{P}_0^{(1)}, \ldots, \mathbb{P}_0^{(T)}$
  dynamic_program($\mathbb{P}_0^{(1)}, \ldots, \mathbb{P}_0^{(T)}$)
  set γ equal to 0
  do
    increment γ
    for all periods $t = 1, \ldots, T$
      crossover, **mutation**, **immigration_from**($t-1$), **immigration_from**(t) and **immigration_from**($t+1$)
    endfor
    dynamic_program($\mathbb{P}_\gamma^{(1)}, \ldots, \mathbb{P}_\gamma^{(T)}$)
  while termination condition is not satisfied

The algorithm terminates if there has not been an improvement during the last N^{nc} generations or if the average of the objective values has not changed more than M^{ch} or if γ exceeds M^{ge}.

4. Results

4.1. Preparations

In order to compare our algorithm to previous results let us consider the two examples P6 and P12 from Yang and Peters [19]. We refer to Yang and Peters [19] for the dimensions, the initial layout and the weight matrix. The weights for moving where all set to 100. As floor area we chose a 30×30 and 50×50 workshop floor, respectively. As the MIP with six departments can be solved completely, we implemented the method described by Yang and Peters [19] replacing the adjacency graph method by a MIP solver. As solution we obtained a single layout for all 6 periods (total cost 6924 about 9% less than in Yang and Peters [19]) which is possible as the adjacency graph method might miss optimal layouts. In order to facilitate the rearrangement we reduced the weights $w_i^{(t)}$ to 19. Depending on the position of the initial layout inside the floor area we obtained two solutions with marginally different total costs. In the centered case the total cost is 6 629 (4 rearrangements) while in the case where we start from a layout in a corner the total cost is 6 634.5 (11 rearrangements). Replacing the adjacency graph method by a GA we obtained for P12 again a single solution for all 4 periods (total cost 28 087.5, 10 rearrangements, about 7% less than in Yang and Peters [19]). That is the reason why we reduced $w_i^{(t)}$ to 50 for this problem (which corresponds to P12A in Yang and Peters [19]). In a single try (as GAs are stochastic methods outcomes of different runs are not identical) we obtained layouts with a total cost of 28 244.5 and 22 rearrangements (about 5% less than in Yang and Peters [19]).

4.2. Dynamic GA

For P6 rearrangement cost $w_i^{(t)}$ will be 19 and in problem P12 the weight $w_i^{(t)}$ is set to 50. As randomness is involved in GAs we run each problem 20 times and evaluated the average, standard deviation, minimum and maximum of the total cost and computation time, see *Table 1*.

	P6	P12		P6	P12
aver. objective	6 569	27 748	average time	29 min 24 s	2 h 40 min
deviation	23.5	344.6	deviation	11 min 20 s	1 h 17 min
worst case	6 613	28 344.5	longest run	57 min 52 s	5 h 14 min
best value	6 507.5	27 098.5	fastest run	8 min 24 s	54 min

Table 1. *Computational results of 20 runs*

All experiments were carried out on a PC with Pentium IV, 1.5 GHz, and we used CPLEX 7.0 for the reduced MIPs. In both cases we started from the initial layout placed in the center of the floor area. The parameters of the GA described above have been set as follows: $M^{\mathsf{ge}} = 100$, $N^{\mathsf{cr}} = 20$, $N^{\mathsf{mu}} = 4$, $N^{\mathsf{im}-} = N^{\mathsf{im}} = N^{\mathsf{im}+} = 2$ (which gives a size of $N^{(t)} = 50$ for each population), the mutation rate was 10%, the probability of accepting individuals worse than average was set to 30%, $N^{\mathsf{nc}} = 20$ and $M^{\mathsf{ch}} = 0.1\%$. Observe that as all crossover and mutation operations are independent of each other one can parallelize this part of the algorithm. The population size $N^{(t)}$ is an important parameter as it determines the exploration of the space of layouts $\mathbb{L}^{(t)}$ and the complexity of the DP for the evaluation of the fitness.

The DP finds for each layout the best rearrangement sequence out of the $\prod_{t=1}^{T} N^{(t)}$ possible ones for which it needs $\sum_{t=0}^{T-1} N^{(t)} N^{(t+1)}$ steps.

In *Figure 1* we summarize graphically the results for P6. Even the solution with the largest total cost 6 613 is better than 6 629 as obtained above. The best run with total cost of 6 507.5 yields an improvement of about 1.8%. On average the computation took less than 30 minutes. The best sequence of layouts is shown in *Figure 2*. *Figure 3* summarizes graphically our experimental results for problem P12. A single outcome is 0.35% worse than 28 244.5 which we obtained above. By contrast, the best result 27 098.5 is about 4% better. On average the computation took less than 2 h 40 min. We have to consider that the number of binary variables in the reduced MIPs increases linearly (which could theoretically yield an exponential increase in computation time). *Figure 4* shows the best sequence of layouts found for P12.

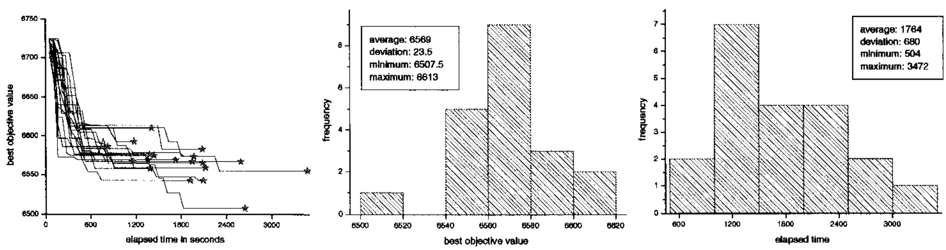

Figure 1. *Diagrams showing the convergence of our dynamic GA for P6, the distribution of the objective values and the computation time for 20 runs*

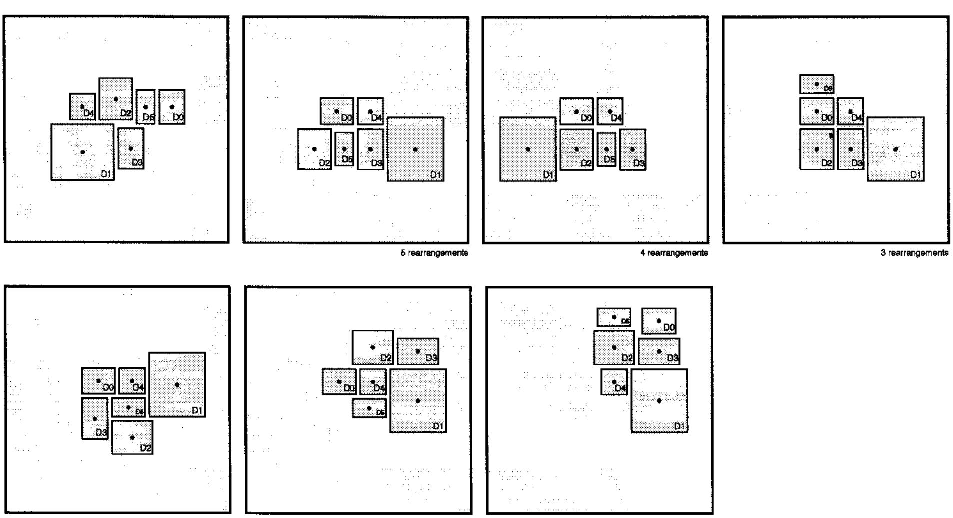

Figure 2. *P6 best layouts found for the six periods with objective value* 6 507.5

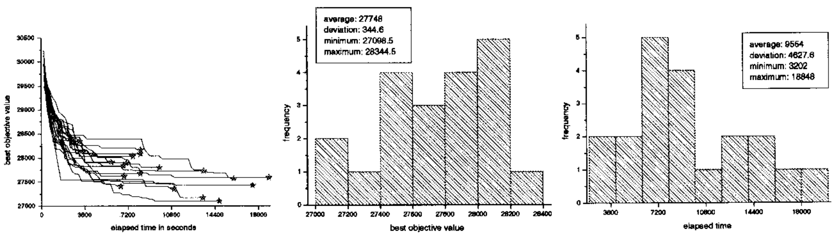

Figure 3. *Diagrams showing the convergence of our dynamic GA for P12, the distribution of the objective values and the computation time for 20 runs*

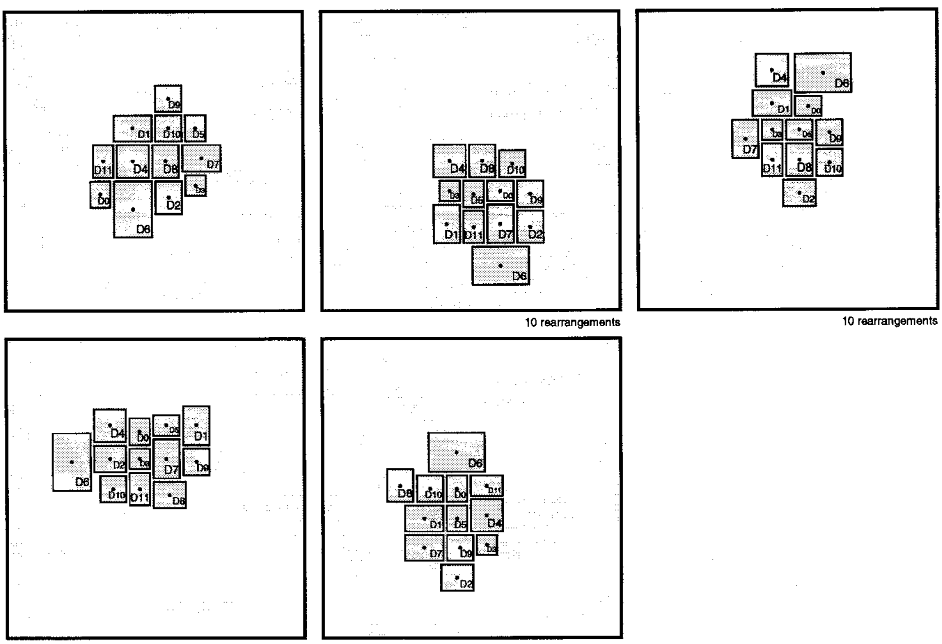

Figure 4. *P12 best layouts found for the four periods with objective value* 27 098.5

5. Conclusion

Concluding we can summarize that our model allows to model dynamic facility layout problems with departments of a different size which can change from period to period. For solving this model we present an algorithm which combines GS and DP. Its search space is larger than the one of the method proposed by Yang and Peters [19]. We use their two examples in order to evaluate the quality of our algorithm. The returned layouts improve results obtained by Yang and Peters [19].

References

[1] J. BALAKRISHNAN, C. H. CHENG, *Dynamic layout algorithms: a state-of-art-survey*, Omega **26**(1998), 507–521.

[2] J. BALAKRISHNAN, C. H. CHENG, *Genetic search and the dynamic layout problem*, Computers and Operation Research **27**(2000), 587–593.

[3] J. BALAKRISHNAN, F. R. JACOBS, M. A. VENKATARAMANAN, *Solutions for the constrained dynamic facility layout problem*, European Journal of Operational Research **57**(1992), 280–286.

[4] J. BALAKRISHNAN, C. H. CHENG, D. G. CONWAY, *An improved pair-wise exchange heuristic for the dynamic plant layout problem*, European Journal of Production Research **38**(2000), 3067–3077.

[5] A. BAYKASOGLU, N. N. Z. GINDY,*A simulated annealing algorithm for dynamic layout problem*, Computers and Operation Research **28**(2001), 1403–1426.

[6] K. C. CHAN, H. TANSRI, *A study of genetic crossover operations of the facilities layout problem*, Computers and Industrial Engineering **26**(1994), 537–550.

[7] D. W. COIT, A. E. SMITH, D. M. TATE, *Adaptive penalty methods for generic optimization of constrained combinatorial problems*, INFORMS Journal of computing **8**(1996), 173–182.

[8] D. G. CONWAY, M. A. VENKATARAMANAN, *Genetic search and the dynamic facility layout problem*, Computers and Operation Research **21**(1994), 955–960.

[9] M. GOETSCHALCKX, *An interactive layout heuristic based on hexagonal adjacency graphs*, European Journal of Operational Research **63**(1992), 304–321.

[10] N. HIRABAYSHI, H. KITA, H. NAGASAWA, *Dynamic facility layout using evolition strategies*, in: Proceedings of the Second World Manufacturing Congress WMC (S. Nahavandi, M. Saadat, Eds.), 1999, 154–159.

[11] B. K. KAKU, J. B. MAZZOLA, *A tabu-search theoristic for the dynamic plant layout problem*, INFORMERS Journal on Computing **9**(1997), 374–384.

[12] J. S. KOCHHAR, S. S. HERAGU, *Facility layout design in a changing enviroment*, International Journal of Production Research **37**(1999), 2429–2446.

[13] T. A. LOCKSONEN, *Preprocessing for static and dynamic facility layout problems*, International Journal of Production Research **35**(1997), 1095–1106.

[14] B. MONTREUIL, A. LAFORGE, *Dynamic layout design given a scenario tree of probable futures*, European Journal of Operational Research **63**(1992), 271–286.

[15] B. MONTREUIL, U. VENKATADRI, *Strategic interpolative design of dynamic manufacturing systems layout*, Management Science **37**(1991), 682–694.

[16] J. MOTTE, I. MURTHY, D. L. OLSON, *A parametric approach to solving bicriterion shortest path problems*, European Journal of Operational Research **53**(1991), 81–92.

[17] M. J. ROSENBLATT, *The dynamics of plant layout*, Management Science **32**(1986), 76–86.

[18] T. L. URBAN, *Solution procedures for the dynamic facility layout problem*, Annals of Operational Research **76**(1998), 323–342.

[19] T. YANG, B. A. PETERS, *Flexible machine layout design for dynamic and uncertain production environments*, European Journal of Operational Research **108**(1998), 49–64.