

UDC 519.859

DOI: 10.32626/2308-5916.2019-19.126-131

**A. Pankratov**, Dr. Sci.,**T. Romanova**, Dr. Sci.

Institute for Mechanical Engineering Problems

National Academy of Sciences of Ukraine, Kharkiv

## DECOMPOSITION ALGORITHM FOR OPTIMIZATION PLACEMENT PROBLEMS

The paper considers a placement problem of 2D convex objects in a rectangular domain of minimum area, that related to the field of Packing and Cutting problems. Our objects may be continuously translated and rotated. A nonlinear programming model of the problem is derived using the phi-function technique. We develop an efficient decomposition algorithm to search for local optimal solutions for the placement problem. The algorithm reduces our problem to a sequence of nonlinear programming subproblems of considerably smaller dimension and a smaller number of nonlinear inequalities. The benefit of this approach is borne out by the computational results.

**Key words:** *placement problem, mathematical model, nonlinear optimization, decomposition algorithm.*

**Introduction.** Optimal placement problem is a part of operational research and computational geometry. It is also known as Packing and Cutting problem [1, p. 1109–1130, 2, p. 397–415]. It has multiple applications in modern biology, mineralogy, medicine, materials science, nanotechnology, robotics, coding, pattern recognition systems, control systems, space apparatus control systems, as well as in the chemical industry, power engineering, mechanical engineering, shipbuilding, aircraft construction, civil engineering, etc. The problems are NP-hard [3, p. 139–183] and, as a result, solution methodologies generally employ heuristics. Some researchers develop approaches based on mathematical modeling and general optimization procedures.

Our approach is based on mathematical modeling of relations between geometric objects, using the phi-function technique (see e. g. [4, p. 539–544, 5, p. 283–294]) and thus reducing the Packing and Cutting problem to a nonlinear programming problem. It contains all globally optimal solutions. It is possible, at least in theory, to use a global solver for the nonlinear programming problem and obtain a solution, which is an optimal packing. However in practice, the model contains a large number of variables and a huge number of inequalities. Specifically, the model involves  $O(n^2)$  nonlinear inequalities and  $O(n^2)$  variables due to additional variables in quasi-phi-functions, where  $n$  is the number of convex objects.

As a result, even finding a locally optimal solution becomes an unrealistic task for the available state of the art NLP-solvers. In order to search for a «good» locally optimal object placement within a reasonable computational time we propose here a decomposition algorithm.

**Problem formulation.** We consider here a placement problem in the following setting. Let  $\Omega$  denote a rectangular domain of length  $l$  and width  $w$ . Both of these dimensions may be variable, or one may be fixed and the other variable. Suppose a set of convex objects  $E_i$ ,  $i \in \{1, 2, \dots, n\} = I_n$ , is given to be placed in  $\Omega$  without overlaps. The position of object  $E_i$  in the fixed coordinates is specified by the coordinates  $(x_i, y_i)$  of its center and the rotation angle  $\theta_i$ . We call  $(x_i, y_i, \theta_i)$  the vector of placement parameters of  $E_i$ . Minimum allowable distances between objects  $E_i$  and  $E_j$ ,  $j > i \in I_n$ , as well as, between each object  $E_i$ ,  $i \in I_n$ , and the frontier (border) of  $\Omega$  may be given.

*Object placement optimization problem.* Place the set of objects  $E_i$ ,  $i \in I_n$ , within a domain  $\Omega = \{(x, y) \in R^2 : 0 \leq x \leq l, 0 \leq y \leq w\}$  of minimum area taking into account distance constraints. If one of the two dimensions ( $l$  or  $w$ ) is fixed, we need to minimize the other one. If both are variable, it is natural to minimize the area  $F = l \cdot w$  of the container.

**Mathematical model.** The vector  $u \in R^\sigma$  of all our variables can be described as follows:  $u = (l, w, u_1, u_2, \dots, u_n, \tau)$ ,  $u_i = (x_i, y_i, \theta_i)$  is the vector of placement parameters for the object  $E_i$ ,  $i \in I_n$ ,  $\tau$  denotes the vector of additional variables, that includes two auxiliary variables  $(\tau_{ij}^1, \tau_{ij}^2)$  for each quasi phi-functions of objects  $E_i$  and  $E_j$ ,  $R^\sigma$  denotes the  $\sigma$ -dimensional Euclidean space, where  $\sigma = 2n^2 + n + 2$ .

A mathematical model of the object placement optimization problem may now be stated in the form:

$$\min_{u \in W \subset R^\sigma} F(u), \quad (1)$$

$$W = \{u \in R^\sigma : \hat{\Phi}_{ij}^1 \geq 0, \hat{\Phi}_i \geq 0, i = 1, 2, \dots, n, j = 1, 2, \dots, n, j > i\}, \quad (2)$$

where  $F(u) = l \cdot w$ ,  $\hat{\Phi}_{ij}^1$  is an adjusted quasi phi-function [5, p. 283–294] defined for the pair of objects  $E_i$  and  $E_j$ ,  $\hat{\Phi}_i$  is an adjusted phi-function [4, p. 539–544] defined for the object  $E_i$  and the object  $\Omega^*$  (to hold the *containment* constraint), taking into account minimum allowable distance  $\rho$ .

Our constrained optimization problem (1), (2) is a continuous nonlinear programming problem. The frontier of  $W$  is usually made of nonlinear surfaces containing valleys, ravines. A matrix of the inequality system which specifies  $W$  is strongly sparse and has a block structure.

Problem (1), (2) is an exact formulation for the object placement optimization problem. Our objective function is a quadratic; each quasi-phi-function inequality in (2) is described by a system of inequalities with differentiable functions.

**A solution strategy.** Our solution strategy consists of three major stages. First we generate a number of starting points from the feasible set of the problem (1), (2). Then starting from each point obtained at Stage 1 we search for a local minimum of the objective function  $F(u)$  of problem (1), (2). Lastly, we choose the best local minimum from those found at Stage 2. This is our best approximation to the global solution of the problem (1), (2).

An essential part of our local optimization scheme (Stage 2) is the decomposition algorithm that reduces the dimension of the problem and computational time. It is due to this reduction that our strategy can process large sets of non-identical convex objects (100 and more). The reduction scheme used by our algorithm is described below. The actual search for a local minimum is performed by IPOPT [6, p. 25–57], which is available at an open access non-commercial software depository (<https://projects.coin-or.org/Ipopt>).

**Description of the Decomposition Algorithm.** Let  $u^{(0)} \in W$  be one of the starting points found by the previous method. The main idea of the algorithm is as follows.

First we circumscribe a circle  $C_i$  of radius  $a_i$  around each object  $E_i$ ,  $i = 1, 2, \dots, n$ . Then for each circle  $C_i$  we construct an «individual» rectangular container  $\Omega_i \supset C_i \supset E_i$  with equal half-sides of length  $a_i + \varepsilon$ ,  $i = 1, 2, \dots, n$ , so that  $C_i$ ,  $E_i$  and  $\Omega_i$  have the same center  $(x_i^0, y_i^0)$  subject to the sides of  $\Omega_i$  being parallel to those of  $\Omega$ ,  $a_i$  is a diameter of  $E_i$ . We take

the fixed value of  $\varepsilon$  of the procedure as  $\varepsilon = \sum_{i=1}^n a_i / n$ . Further we fix the position of each individual container  $\Omega_i$  and let the local optimization algorithm

move the corresponding object  $E_i$  only within the container  $\Omega_i$ . It is clear that if two individual containers  $\Omega_i$  and  $\Omega_j$  do not have common interior points for  $\rho = 0$ , i. e.  $\Phi^{\Omega_i, \Omega_j} \geq 0$ , (or  $\text{dist}(\Omega_i, \Omega_j) \geq \rho$  for  $\rho > 0$ , i. e.  $\widehat{\Phi}^{\Omega_i, \Omega_j} \geq 0$ ), then we do not need to check the non-overlapping (or distance) constraint for the corresponding pair of objects  $E_i$  and  $E_j$ .

The above key idea allows us to extract subsets of our feasible set  $W$  of the problem (1), (2) at each step of our optimization algorithm as follows.

We create an inequality system of additional constraints on the translation vector  $v_i$  of each object  $E_i$  in the form:  $\Phi^{C_i\Omega_{1i}^*} \geq 0$ ,  $i \in I_n$ , where  $\Phi^{C_i\Omega_{1i}^*} = \min\{-x_i + x_i^0 + \varepsilon, -y_i + y_i^0 + \varepsilon, x_i - x_i^0 + \varepsilon, y_i - y_i^0 + \varepsilon\}$ , is the phi-function for the circle  $C_i$  and object  $\Omega_{1i}^* = R^2 \setminus \text{int } \Omega_{1i}$ .

The inequality  $\Phi^{C_i\Omega_{1i}^*} \geq 0$  is equivalent to the system of four linear inequalities  $-x_i + x_i^0 + \varepsilon \geq 0$ ,  $-y_i + y_i^0 + \varepsilon \geq 0$ ,  $x_i - x_i^0 + \varepsilon \geq 0$ ,  $y_i - y_i^0 + \varepsilon \geq 0$ .

Then we form a new subregion defined by

$$W_1 = \{u \in R^{\sigma-\sigma_1} : \widehat{\Phi}_{ij} \geq 0, (i, j) \in \Xi_1, \widehat{\Phi}_i \geq 0, \Phi^{C_i\Omega_{1i}^*} \geq 0, i \in I_n\},$$

where  $\Xi_1 = \{(i, j) : \widehat{\Phi}^{\Omega_{1i}\Omega_{1j}} < 0, i > j = 1, 2, \dots, n\}$ .

In other words, we delete from the system, which describes feasible region  $W$ , quasi phi-function inequalities for all pairs of objects whose individual containers do not overlap and we add additional inequalities  $\Phi^{C_i\Omega_{1i}^*} \geq 0$ , which describe the containment of the circles  $C_i$  in their individual containers  $\Omega_{1i}$ ,  $i = 1, 2, \dots, n$ . Eo ipso we reduce the number of additional variables by  $\sigma_1$ . Then our algorithm searches for a point of local minimum  $u_{w_1}^*$  of the subproblem

$$\min_{u_{w_1} \in W_1 \subset R^{\sigma-\sigma_1}} F(u_{w_1}).$$

When the point  $u_{w_1}^*$  is found, it is used to construct a starting point  $u^{(1)}$  for the second iteration of our optimization procedure.

At that iteration we again identify all the pairs of objects with non-overlapping individual containers, form the corresponding subregion  $W_2$  (analogously to  $W_1$ ) and let our algorithm search for a local minimum  $u_{w_2}^* \in W_2$ . The resulting local minimum  $u_{w_2}^*$  is used to construct a starting point  $u^{(2)}$  for the third iteration, etc.

Then we solve the  $k$ -th subproblem with starting point  $u^{(k-1)}$  on a subregion  $W_k$ :

$$\min_{u_{w_k} \in W_k \subset R^{\sigma-\sigma_k}} F(u_{w_k}), \tag{3}$$

$$W_k = \{u \in R^{\sigma - \sigma_k} : \widehat{\Phi}'_{ij} \geq 0, (i, j) \in \Xi_k, \widehat{\Phi}_i \geq 0, \Phi^{C, \Omega_i} \geq 0, i \in I_n\}, \quad (4)$$

where  $\Xi_k = \{(i, j) : \widehat{\Phi}^{\Omega_u \Omega_v} < 0, i > j = 1, 2, \dots, n\}$ .

If the point  $u_{w_k}^*$  of local minimum of the  $k$ -th subproblem belongs to the frontier of an «artificial» subset

$$\Pi_k^\varepsilon = \{u \in R^{\sigma - \sigma_k} : -x_i + x_i^{(k-1)} + \varepsilon \geq 0, -y_i + y_i^{(k-1)} + \varepsilon \geq 0, \\ x_i - x_i^{(k-1)} + \varepsilon \geq 0, y_i - y_i^{(k-1)} + \varepsilon \geq 0, i = 1, \dots, n\},$$

(i. e.  $u_{w_k}^* \in fr \Pi_k^\varepsilon$ ), we take the point  $u_{w_k}^* = u^{(k)}$  as a center point for a new subset  $\Pi_{k+1}^\varepsilon$  and continue our optimization procedure, otherwise (i. e.  $u_{w_k}^* \in int \Pi_k^\varepsilon$ ) we stop our iterative procedure.

We note that  $\text{dist}(u_{w_k}^*, u_{w_{k+1}}^*) \geq \varepsilon$ , if  $u_{w_{k+1}}^* \in fr \Pi_k^\varepsilon$ , and the value of  $\varepsilon$  is considerably greater than the accuracy of IPOPT ( $10^{-8}$ ). Thus, we may conclude that the stopping condition of the decomposition algorithm is always reached in a finite number of iterations.

We claim that the point  $u^* = u^{(k)*} = (u_{w_k}^*, \tau_k^*) \in R^\sigma$  is a point of local minimum of the problem (1), (2), where  $u_{w_k}^* \in R^{\sigma - \sigma_k}$  is the last point of our iterative procedure and  $\tau_k^*$  is a vector of redefined values of the previously deleted additional variables  $\tau_k \in R^{\sigma_k}$ . The assertion comes from the fact that any arrangement of each pair of objects  $E_i$  and  $E_j$  subject to  $(i, j) \in \Xi \setminus \Xi_k$  guarantees that there always exists a vector  $\tau_k$  of additional variables such that  $\widehat{\Phi}'_{ij} \geq 0, (i, j) \in \Xi \setminus \Xi_k$  at the point  $u^{(k)*}$ . Here  $\Xi = \{(i, j), i > j = 1, 2, \dots, n\}$ . Therefore the values of additional variables of the vector  $\tau_k$  have no effect on the value of our objective function, i. e.  $F(u_{w_k}^*) = F(u^{(k)*})$ . That is why, indeed, we do not need to redefine the deleted additional variables of the vector  $\tau_k$  at the last step of our algorithm.

So, while there are  $O(n^2)$  pairs of objects in the container, our algorithm may in most cases only actively controls  $O(n)$  pairs of objects (this depends on the sizes of objects and the value of  $\varepsilon$ ), because for each object only its « $\varepsilon$ -neighbors» have to be monitored.

The parameter  $\varepsilon$  provides a balance between the number of inequalities in each nonlinear programming subproblem and the number of the subproblems which we need to generate and solve in order to get a local optimal solution of problem (1), (2).

**Concluding remarks.** The proposed decomposition algorithm allows us to reduce the problem (1), (2) with  $O(n^2)$  inequalities and a  $O(n^2)$ -dimensional feasible set  $W$  to a sequence of subproblems (3), (4), each with  $O(n)$  inequalities and a  $O(n)$ -dimensional solution subset  $W_k$ . This reduction is of a paramount importance, since we deal with nonlinear optimization problems. We are going to apply our algorithm to optimization placement problems for composed 2D and 3D objects in the near future.

### Reference:

1. Wascher G., Hauner H. and Schumann H. An improved typology of cutting and packing problems. *European Journal of Operational Research*. 2007. Vol. 183 (3), № 16. P. 1109–1130.
2. Bennell J., Oliveira J. The geometry of nesting problems. A tutorial. *European J. Operational Research*. 2008. Vol. 184. P. 397–415.
3. Chazelle B., Edelsbrunner H., Guibas L. The complexity of cutting complexes. *Discrete & Computational Geometry*. 1989. Vol. 4 (2). P. 139–181.
4. Chernov N., Stoyan Yu., Romanova T. Mathematical model and efficient algorithms for object packing problem. *Computational Geometry: Theory and Applications*. 2010. Vol. 43 (5). P. 535–553.
5. Stoyan Yu., Pankratov A., Romanova T. Quasi-phi-functions and optimal packing of ellipses. *Journal of Global Optimization*. 2016. Vol. 65 (2). P. 283–307.
6. Wachter A., Biegler L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*. 2006. Vol. 106 (1). P. 25–57.

## АЛГОРИТМ ДЕКОМПОЗИЦІЇ ДЛЯ РОЗВ'ЯЗАННЯ ОПТИМІЗАЦІЙНИХ ЗАДАЧ РОЗМІЩЕННЯ

У статті розглядається задача розміщення двовимірних опуклих об'єктів у прямокутній області мінімальної площі, яка відноситься до класу задач упаковки і розкрою. Об'єкти, що розміщуються, можуть неперервно транслюватися і обертатися. Будується математична модель задачі розміщення у вигляді задачі нелінійного програмування з використанням методу  $\phi$ -функцій. Для пошуку локально-оптимальних розв'язків пропонується ефективний алгоритм декомпозиції, який зводить вихідну задачу до послідовності підзадач нелінійного програмування значно меншою розмірності з меншим числом нелінійних нерівностей. Перевага цього підходу підтверджується результатами численних експериментів.

**Ключові слова:** задача розміщення, математична модель, нелінійна оптимізація, алгоритм декомпозиції.

Date received 31.01.2019