Second International Conference on Construction in the 21st Century (CITC-II)
"Sustainability and Innovation in Management and Technology"
10-12 December, 2003, Hong Kong

A Reactive Greedy Randomized Adaptive Search Procedure **for Construction-Site Layout**

K. P. Anagnostopoulos
Assistant Professor, Democritus University of Thrace, 671 00 Xanthi, Greece

Abstract
Construction-site layout is to determine what temporary facilities are required to support construction activities and to position them in appropriate locations. A site-level facility layout has an important impact on the production time and cost savings, especially for large projects, as well as on safety of operations and on environmental aspects of the work. The construction-site layout problem is formulated as a location-allocation problem. The objective is to determine an assignment of facilities to locations in order to minimize the total cost, i.e., the sum of the total construction and removal cost of assigning a facility on a location, the total transportation cost of materials and the transportation of personnel between facilities and locations. The problem is solved by a Greedy Randomized Adaptive Search Procedure (GRASP), enhanced by a learning mechanism and a bias function for determine the next element to be introduced in the solution. The procedure has been coded in Visual Basic, and computational results are given.

Keywords
GRASP, Construction-Site Layout, Quadratic Assignment

1. Introduction

Construction site-layout involves coordinating the use of limited space to accommodate temporary facilities so that they can function efficiently on site. Temporary facilities (such as warehouses, fabrication shops, maintenance shops, batch plants, and residence facilities) have the general function to support construction activities. Good site-layout planning is crucial for enhancing the productivity on construction sites, since low productivity is highly linked to inefficient space planning resulting in increased transportation costs and nonproductive time, as well as in the inefficient use of resources and conflicts between subcontractors.

The layout problem is generally defined as the problem of identifying the shape and size of the facilities to be laid out, and determining the relative positions of these facilities that satisfy the constraints between them and allow them to function efficiently (Zouein et al, 2002). Various classes of layout problems have been studied in the literature. In some studies, the layout problem was modeled as a location-allocation problem, which consists of allocating a set of predetermined facilities into a set of predetermined sites where the smallest site can accommodate the largest facility (Li and Love, 1998; Yeh, 1995). In other models, a facility can take any user-specified shape (Hegazy and Elbeltagi 1999); two-dimensional geometric constraints between facilities are considered (Zouein et al, 2002); one facility at most can be set up on specific area to avoid overlapping and some facilities can not be positioned within a given distance of each other (Mawdesley et al, 2002). Construction-layout planning can also be dynamic, i.e. layouts that change over time as construction progresses (Zouein and Tommelein, 1999) or can be formulated as a multicriteria-multiobjective problem (Tam et al, 2002).

Many solutions methods have been proposed in the literature for solving the site-layout planning problem. Since the layout problem is an NP-hard combinatorial problem, optimization-based procedures are limited to small size instances and researchers often resort to using heuristics to find acceptable solutions, or hybrid methods combining heuristics and exact algorithms (Zouein and Tommelein 1999). In recent years, new techniques and metaheuristics, especially genetic algorithms, were mainly proposed. I. C. Yeh (1995) has used artificial neural networks to minimize a total cost function that includes the cost of constructing a facility at the assigned location on site and the cost of interacting with other facilities. Genetic algorithms have also been applied to solve the layout-improvement problem and the geometrically constrained site layout problem (Hegazy and Elbeltagi, 1999; Li and Love, 1998; Mawdesley et al, 2002; Zouein et al, 2002).

This paper presents a *Greedy Randomized Adaptive Search Procedure* (GRASP) for solving the site-layout problem. The standard GRASP was enhanced by a learning mechanism and a bias function for determine the next element to be introduced in the solution. The procedure has been coded in Visual Basic, and it was tested on randomly generated problems in order to find out suitable values of its parameters.

2. Construction Site-Level Layout Problem

In the present work, the formulation of the construction-site layout problem is that a set of n predetermined facilities needs to be located on n predetermined locations on the site. Each facility should be assigned on one and only one site, and each site should be assigned with one and only one facility. We assume that each of the predetermined locations is capable of accommodating the largest one among the facilities. The construction site-level layout problem to be solved can be stated as follows:

$$\text{minimize} TC = \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ik} + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} \sum_{r=1}^{m} p_{ijr} q_{ijr} d_{kl} x_{ik} x_{jl}$$
$$+ \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} u_{ij} f_{ij} d_{kl} x_{ik} x_{jl}$$

subject to

$$\sum_{i=1}^{n} x_{ik} = 1 \qquad k = 1,\dots,n$$

$$\sum_{k=1}^{n} x_{ik} = 1 \qquad i = 1,\dots,n$$

$$x_{ik} = 0-1 \qquad i,k = 1,\dots,n$$

where

x_{ik} = 1 if facility i is assigned on site k; 0 otherwise.
c_{ik} = construction and removal cost of assigning facility i on site k.
d_{kl} = distance between location k and l; q_{ijr} quantity of material r transported from facility i to facility j.
p_{ijr} = transport price of unit quantity of material r from facility i to j.
f_{ij} = frequency trips of personnel from facility i to facility j; u_{ij} the unitary cost of a personnel trip from facility i to j.

The objective of site-level facility layout is to find an assignment of facilities on locations in order to minimize the total cost *TC*, in which the first term accounts for the total construction and removal cost of assigning a facility on a site; the second term represents the total cost of transporting materials between facilities; and the third term represents the total cost of the site personnel traveling between facilities.

No geometric or others constraints are imposed on the relative positions of facilities. Despite this simplification, we can recognize under this formulation one of the most difficult combinatorial problems, namely the quadratic assignment problem, of which the well-known traveling salesman problem is a special case (Li et al, 1994).

```
procedure GRASP(maxiter)
    BestCost ← ∝;
    for i = 1, …, maxiter do
        s^current ← GreedyConstruction();
        s^current ← LocalSearch(s^current);
        if c(s^current) < BestCost do
            s^best ← s^current; BestCost ← c(s^current);
        end if;
    end for;
    return s^best;
end GRASP;
```

Figure 1: The pseudo-codes of the standard GRASP procedure

3. Construction Site-level Layout using Reactive GRASP

The *Greedy Randomized Adaptive Search Procedure* (GRASP) is a metaheuristic for solving hard combinatorial optimization problems. GRASP is an iterative process, in which each iteration consists of two phases: construction and local search (Feo and Resende, 1995; Resende and Ribeiro, 2002). The construction phase builds a feasible solution $s^{current}$, whose neighborhood is explored by local search. The best overall solution s^{best} is kept as the result (fig. 1). Li Y. et al (1994) have proposed a standard GRASP for the quadratic assignment problem,

3.1 The Construction Phase

In the construction phase, a feasible solution is created step by step (fig. 2). At each construction iteration, the next assignment (i, k) to be added is determined by ordering the available candidates in a sorted list, called *Restricted Candidate List* (RCL), with respect to a greedy function that measures the benefit of selecting each assignment.

The construction phase begins by assigning at random a facility to a location. We denote by c(i, k) the incremental cost associated with the incorporation of a new assignment into the solution under construction with respect to the already-made assignments, i.e. we select from the facility-location pairs not already assigned, the one that has the minimum cost as calculated by the objective function TC. At any GRASP iteration, let c_{min} and c_{max} be, respectively, the smallest and the largest incremental costs. The RCL is formed by all assignments whose quality is superior to a threshold value, i.e., $c_{min} \le c(i, k) \le c_{min} + \alpha(c_{max} - c_{min})$, where $0 \le \alpha \le 1$. The cases $\alpha = 0$ and $\alpha = 1$ correspond to a pure greedy algorithm and to a random construction respectively.

The standard GRASP does not use information from the solutions found in previous iterations. In the Reactive GRASP, developed by Prais and Ribeiro (2000), the value of α is selected at each iteration from a discrete set of possible values. The solution values found along the previous iterations guide this selection. Let T = $\{\alpha_1, \ldots, \alpha_\tau\}$ be the set of possible values for α. The probabilities associated with the choice of each value are all initially made equal to $\pi_i = 1/\tau$ (i = 1,…, τ). Furthermore, let be the incumbent solution and let TC_i be the average value of all solutions found using α_i (i = 1,…, τ). The selection probabilities are periodically reevaluated by taking

The value of q_i will be larger for values of $\alpha = \alpha_i$ leading to the best solutions on average.

```
procedure GreedyConstruction()
    s^current ← ∅; Select at random an assignment (i, k);
    s^current ←(i, k);
    Evaluate the incremental costs of the candidate assignments;
    while s^current is not a complete solution do
        Build the restricted candidate list (RCL);
        Select an assignment (i, k) from the RCL at random;
        Add (i, k) in s^current;
        Reevaluate the incremental costs;
    end while;
    return s^current;
end GreedyConstruction;
```

Figure 2: The pseudo-codes of the construction phase

In the basic GRASP, the next assignment to be introduced in the solution is selected at random from the RCL, all candidate assignments of the RCL having equal probabilities of being chosen. Another selection mechanism is based on the rank assigned to each candidate assignment (i, k), according to its value of the greedy function (Resende and Ribeiro, 2002). Let r_{ik} denote the rank of assignment (i, k), and let bias(r_{ik}) be a bias function (see next section). Once these values have been evaluated for all assignments of the RCL, the probability $\pi(i, k)$ of selecting assignment (i, k) is

$$\pi(i,k) = \frac{bias(r_{ik})}{\sum_{(j,l)\in RCL} bias(r_{jl})}$$

```
procedure LocalSearch(s^current)
    while s^current is not locally optimal do
        Find s′ in the neighborhood of s^current with c(s′) < c(s^current);
        s^current ← s′;
    end while;
    return s^current;
end LocalSearch;
```

Figure 3: The pseudo-code of the GRASP procedure

3.2 The Local Search Phase

Since the solutions generated by a GRASP construction are not guaranteed to be locally optimal, a local search is usually applied to improve each constructed solution (fig. 3). A local search algorithm works in an iterative fashion by successively replacing the current solution by a better solution, if one is found, in the neighborhood of the current solution. It terminates when no better solution is found in the neighborhood. Figure 3 illustrates the local search algorithm starting from the solution $s^{current}$ constructed in the first phase. We use the so-called 2-exchange neighborhood structure, i.e. all solutions for which the only difference to current solution consists of exchanging the locations of two facilities. Once a 2-exchange that improves the current solution is found, the two assignments are fixed and the procedure is repeated using the rest of candidate assignments.

4. Computational Results

The algorithm has been coded in Visual Basic and was tested through Monte Carlo experimentation in order to evaluate the performances according to different values of its parameters. The analysis has been realized on a personal computer with Pentium III processor operating at 1000 MHz. In each case 20 randomly generated instances of the problem have been resolved with sizes ranging from $n = 10$ to $n = 60$ facilities. If the value of a parameter were proved to be the best, it was adopted in the subsequent experiments.

1. In the first experiment the fixed values $\alpha = 25\% - 50\% - 75\%$ were tested against the Reactive GRASP with values of α selected at each iteration from the discrete set T = {10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 100%}. The selection probabilities were reevaluated after 100 iterations (maxiter = 5000). The Reactive GRASP has give better results than those found by the basic GRASP (45%, 50% and 45% respectively) in 65% of cases. The average percentage of iterations until the Reactive GRASP has found out the local optimum solution was 40,46% of maxiter.
2. In the second experiment the following bias functions were tested in order to select the best one (r denotes the rank of an assignment):
   - Random bias: bias(r) = 1(all candidate assignments of the RCL have equal probabilities of being chosen)
   - Linear bias: bias(r) = $1/r$
   - Exponential bias: bias(r) = e^{-r}
   - Polynomial bias of order 2: bias(r) = r^{-2}

   The Reactive GRASP with exponential bias function was proven to be the best in 70% of cases.
3. In order to test the quality of obtained solutions, first, a program solving the problem by complete enumeration of solutions was written and run on 20 instances with n = 10, 11, 12. The Reactive GRASP with exponential bias function has found the best solution in all cases. Second, the average percentage of relative improvement of GRASP over the solution provided by one run of local search procedure was 3,46%.

5. Conclusion

A Greedy Randomized Adaptive Search Procedure (GRASP) was developed and applied to construction-site layout problems with up to fifty facilities. The problem formulation aims at minimizing the total construction and removal cost of facilities, and of the total transportation cost of materials and personnel between facilities and locations. The basic GRASP was enhanced by a learning mechanism and a bias function for determine the next element to be introduced in the solution. Preliminary experiments have been conducted to determine suitable parameter values for the GRASP algorithm. A possible extension of this study is to incorporate the recent techniques of path relinking and reverse path relinking to the initial GRASP, in order to improve its results. This is the subject of current investigation by the writer.

6. References

Feo, T. A., and Resende, M. G. C. (1995). "Greedy randomized adaptive search procedures." *Journal of Global Optimization*, Vol. 6, pp. 109-133.

Hegazy, T., and Elbeltagi, E. (1999). "EVOSITE: Evolution-based model for site layout planning." *Journal of Computing in Civil Engineering*, Vol. 13, No. 3, pp. 198-206.

Li, H., and Love, P. E. D. (1998). "Site-level facilities layout using genetic algorithms." *Journal of Computing in Civil Engineering*, Vol. 12, No. 4, pp. 227–231.

Li, Y., Pardalos, P.M., and Resende, M.G.C. (1994). "A greedy randomized adaptive search procedure for the quadratic assignment problem", *Quadratic Assignment and Related Problems,* Editors: P.M. Pardalos and H. Wolkowicz, Vol. 16 of DIMACS Series on Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, pp. 237-261.

Mawdesley, M. J., Al-jibouri S. H., and Yang, H. (2002). "Genetic algorithms for construction site layout in project planning." *Journal of Construction Engineering and Management*, Vol. 128, No. 5, pp. 418-426.

Prais, M., and Ribeiro, C.C. (2000). "Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment." *INFORMS Journal on Computing*, Vol. 12, pp. 164-176.

Resende, M. G. C., and Ribeiro, C. C. (2002). "Greedy randomized adaptive search procedures." *AT&T Labs Research Technical Report TD-53RSJY, version 2*.

Tam, C. M., Tong, T. K. L., Leung, A. W. T., and Chiu, G. W. C. (2002)."Site layout planning using nonstructural fuzzy decision support system." *Journal of Construction Engineering and Management*, Vol. 128, No. 3, pp. 220-231.

Yeh, I. C. (1995). "Construction-site layout using annealed neutral network." *Journal of Computing in Civil Engineering*, Vol. 9, No. 3, pp. 201–208.

Zouein, P. P., and Tommelein, I. D. (1999). "Dynamic layout planning using a hybrid incremental solution method." *Journal of Construction Engineering and Management,* Vol. 125, No. 6, pp. 400–408.

Zouein, P. P., Harmanani H., and Hajar, A. (2002). "Genetic algorithm for solving site layout problem with unequal-size and constrained facilities." *Journal of Computing in Civil Engineering*, Vol. 16, No. 2, pp. 143-151.